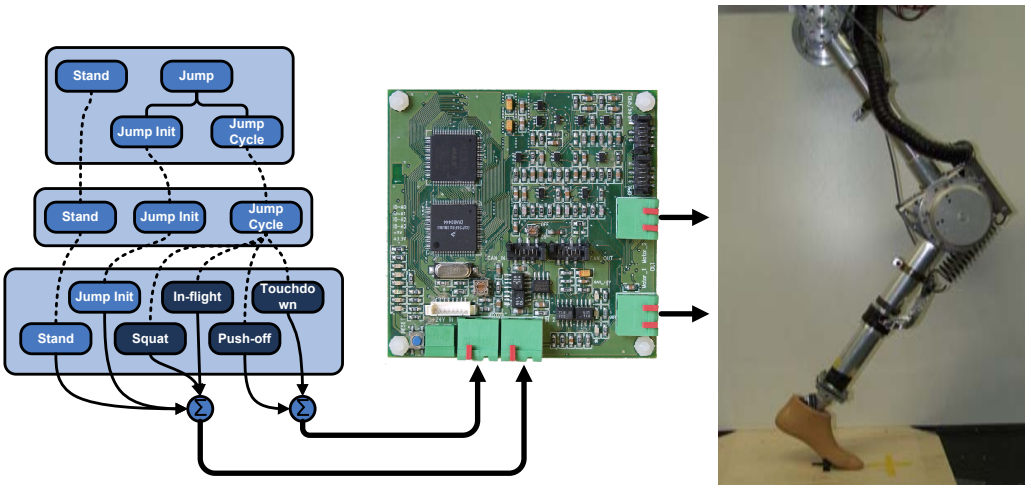
ROBOTICS RESEARCH LAB
DEPARTMENT COMPUTER SCIENCES
UNIVERSITY OF KAISERSLAUTERN

Diploma Thesis



Distributed Compliant Control for Biologically Inspired Behavior-Based Robots

Sebastian Blank

January 14, 2009

Diploma Thesis

Distributed Compliant Control for Biologically Inspired Behavior-Based Robots

Robotics Research Lab
Department of Computer Sciences
UNIVERSITY OF KAISERSLAUTERN

Sebastian Blank

Day of issue : June 20, 2008
Day of release : January 14, 2009

Professor : Prof. Dr. Karsten Berns
Tutor : Dipl.Ing. Thomas Wahl

Hereby I declare that I have self-dependently composed the Diploma Thesis at hand. The sources and additives used have been marked in the text and are exhaustively given in the bibliography.

January 14, 2009 – Kaiserslautern

(Sebastian Blank)

Preface

I would like to take the opportunity to thank the people that supported me in the process of working on this thesis. First of all, I would like to thank my supervision professor Karsten Berns for the opportunity to work on this project at the Robotics Research Lab and his support throughout my years as a student assistant . The second person that I would like to mention is my tutor Thomas Wahl. Without our productive discussions concerning the concept and the countless hours of hands-on assistance, this thesis would not be the same. Among the rest of the helpful staff members, I would like to single out three persons that made outstanding contributions. Those people are Carsten Hillenbrand, Tobias Luksch, and Lothar Gauß. They enriched me with their profound knowledge and helped me with their advise.

Furthermore, I would like to express my gratitude to all people involved in the motion capture experiment for making the time to participate. In details those people are Thomas Jaitner, Daniel Schmidt, Raphael Reitzig, Christiano Gava, Jochen Hirth, and Max Steiner. Moreover, I would like to give a special thank to the people that proof-read large sections of the work at hand: Corinna Chudalla and Reiner Hüchting.

Last but not least I would like to thank my parents for their support throughout all these years and my girlfriend Monika Csemba for living through these past months with me, all the support she gave me with this work, and everything else she has done for me. I would not have been able to finish this thesis without them.

Contents

1. Introduction

1.1 Biologically Inspired Design and Control

In its rather short history robotic research has come a long way in the half century since it started to exist as a noticeable scientific field. Due to its roots in engineering, computer science, mathematics, and several other 'classical' scientific branches, a grand diversity of methodologies and approaches existed from the very beginning. Hence, the researchers in this field are in particular used to adopting ideas that originate in other fields. As a fairly logical consequence of this, scientists tended to biology during the 1970s in order to find approaches that are ideally adapted to the conditions of our natural environment. Doing so allows for introducing principles to robotics that have already shown their great potential by prevailing in a tough evolutionary selection process for millions of years. The variety of these approaches spans from efficient locomotion, to sensor processing methodologies and all the way to control architectures. Thus, the full spectrum of challenges for autonomous interaction with the surroundings while pursuing a task can be covered by such means.

A feature that has proven to be amongst the most challenging to recreate is the human ability of biped locomotion. This is mainly caused by the fact that walking, running and so on are highly complex processes involving the need for energy efficient actuation, sophisticated control architectures and algorithms, and an elaborate mechanical design while at the same time posting restrictions concerning stability and weight. However, it is of special interest since our environment is favoring this specific kind of locomotion and thus promises to open up an enormous potential if mastered. More than the mere scientific interest, it is the fascination of understanding and recreating parts of oneself that drives the ongoing efforts in this area of research.

The fact that this is not at all an easy task to tackle is not only caused by the highly dynamical processes but also has its roots in the challenging design process. That is because it cannot be limited to just one aspect like e.g. the control architecture, actuation, sensors, or mechanical design alone. Each aspect has to be incorporated into a sound general concept in order to allow for a successful outcome in the end.

Since control is in this context inseparably coupled with the mechanics of the system, both has to be dealt with here.

1.2 Motivation

The publicly most well known ambassador from the field of humanoid walking machines is Honda's Asimo[1] (see figure 1.1).

Figure 1.1: ASIMO

At first glance this platform offers the most advanced features of all since it is not only capable of stable walking with a human like pace of approx. 2.7 km/h but also copes with running while reaching a remarkable speed of up to 6 km/h. From a scientific point of view the above is not necessarily true for various reasons:

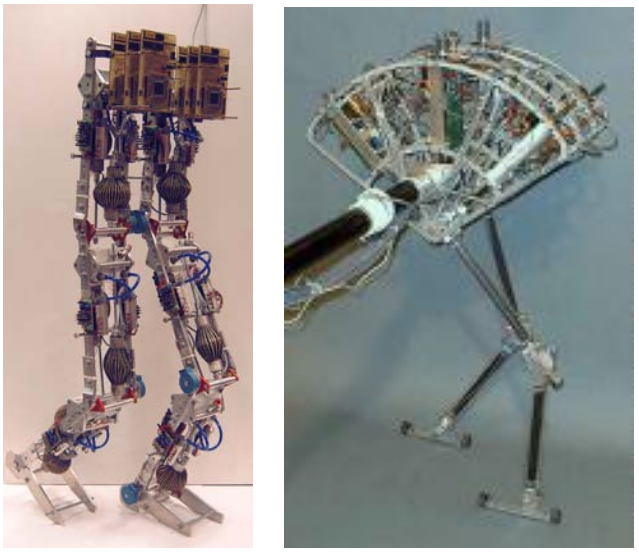
- It is highly energy inefficient since it does not apply biological principles for the locomotion task but rather uses endless amounts of power to imitate human movements. This is based on the trajectory control-law applied and results in a rather poor degree of autonomy.
- It is not able to adapt to disturbances in its environment and thus requires specific lab conditions (perfectly even surface with sufficient friction) to be able to function correctly. Unfortunately this holds true for the robot's ability to e.g. climb stairs that have to meet very narrow specifications concerning dimensions and distance in order to allow it to climb them.
- It is tele-operated instead of being capable of autarkic planning and operation.

In the author's opinion this disqualifies the robot as a source of inspiration for the concept of the project at hand because of the drastic disadvantages that have been

[1] **A**dvanced **S**tep in **I**nnovative **MO**bility, *(jap.: ashimo)* - with legs

pointed out above. Therefore let us redirect our attention to approaches taken by other researchers in order to master human-like locomotion[2].

The first major step towards the biological role models is the ability to use energy efficiently. For this purpose, several groups make use of compliant actuation principles. Outstanding examples of the application of mechanical compliance can be traced back almost two decades. These are SPRING FLAMINGO (figure 1.2(b)) built by the Massachusetts Institute of Technology Leg Laboratory and LUCY (figure 1.2(a)) developed at the University of Brussels, Belgium.



(a) LUCY **(b)** SPRING FLAMINGO

Figure 1.2: Bipedal walking machines making use of compliant actuators developed in the 1990s

The former makes use of springs that are arranged in series with the actuator while the latter relies on fluidic muscles. A more detailed description of the actuators can be found in the following section (2.1). Although the approaches taken by the two groups seem very different they both have impressively demonstrated the potential that compliant actuation has to offer. Therefore more advanced actuators of this kind have evolved over the last decade that were used on the next-generation of biologically inspired and energy efficient walking machines like VERONICA (figure 1.3(a), University of Brussels) and FLAME (figure 1.3(b)) developed at the Technical University Delft, Netherlands.

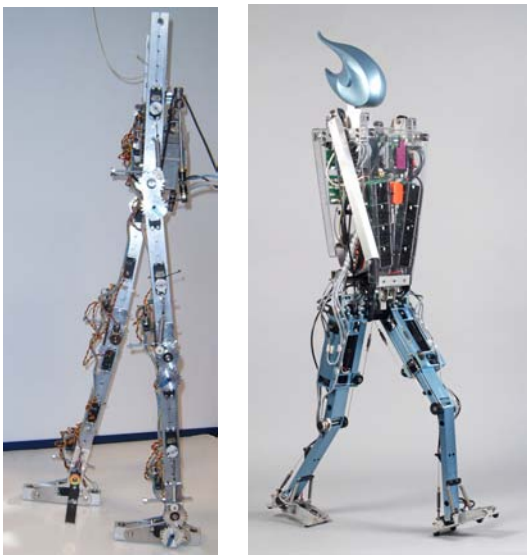
Both approaches seem very promising although very few results have been published so far since both groups just recently started with the work on the respective project.

A general trend in almost all robots intended for human like locomotion is the drastic increase in DOF[3] that can be found recently. A good example is the concept for the robot LOLA [4] (figure 1.4).

[2] An overview of the features of the mentioned robots can be found in the appendix (B.1)

[3] Degree(s) of freedom

[4] Details can be found in [Ulbrich 06] and [Lohmeier 04]



(a) VERONICA **(b)** FLAME

Figure 1.3: Next generation compliant bipedal machines

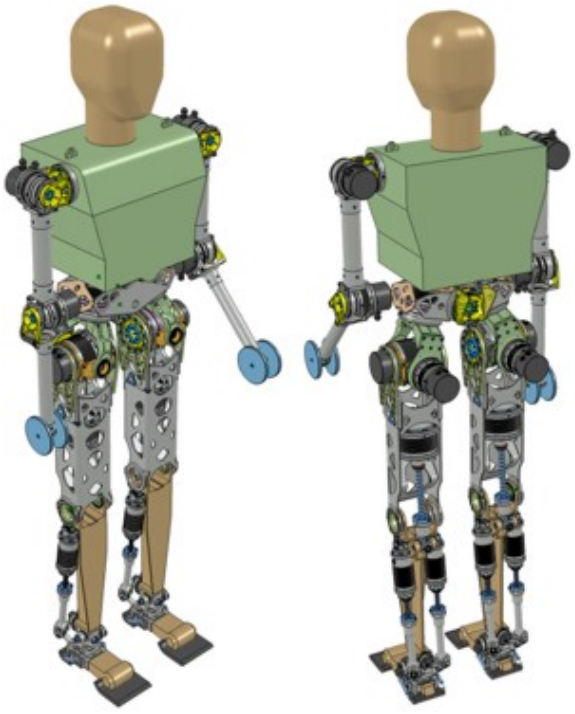


Figure 1.4: LOLA, TU Munich, Germany

Even though no pictures or results of the actual robot have been published yet, the concept reveals a significant increase in the complexity of the kinematic setup that is directly related to the increase in DOF. On the one hand, this is the consequence of the attempt to come closer tho the very sophisticated setup that can be found in a human being. On the other, however, this increases the demand concerning computational power, system latency, robustness, and electrical wiring. A straightforward solution to the arising problem is the introduction of physically distributed components in the control architecture. A good example for this can be found in robots of 'The Humanoid Robotics Project' named HRP-2 and HRP-3 (see figure 1.5).

Figure 1.5: HRP-2

Both make use of distributed control components in order to overcome the challenges mentioned before. Unfortunately this comes with the price of an increased complexity of the control design.

Therefore, one can say that various approaches have been undertaken during the last years in order to increase the performance of bipedal walking machines. Most of them showed great potential while at the same time imposing several drawbacks that will be pointed out in the following chapter. Hence, the idea is to combine several of these features in order to benefit from the advantages and overcome their individual weak spots by introducing a control architecture suited for the jumping task.

1.3 Objectives of this Thesis

A popular opinion among researchers is that there are only two possible ways to go when designing a control system: the classic engineering approach that features feedback control or make use of biological paradigms. This harsh separation however, is not necessarily true, nor is it useful. Thus, the attempt of this thesis is to combine principles that are found in animals with a classical low level control into one system. This approach is extended into the mechanical design as well. Therefore, one is able to benefit from the advantages of both worlds. This results in a system that makes use of inherent dynamics by introduction of elastic mechanical

elements and compliant feedback control of the actuators while applying a biologically inspired hierarchical behavior based control architecture. This ensures natural movements without the need to achieve them by high power actuation. Thus, high energy efficiency can be achieved. The overall task is to provide a system capable of controlling a monopod leg that is intended to perform cyclic jump motions.

The application topic was selected since the process of jumping is ideally suited to be used as a form of benchmark for the system's capabilities. This is because jumping is a highly dynamic process that requires very low latencies in processing the sensor information available and therefore enables the system to come to a reliable and expeditious situation estimation as a basis for making an adequate control decision. For this purpose, a control architecture suited for the scenario must be both fast and robust at the same time. These postulations imply a multi level behavior based control architecture that offers distributed low level control aspects on the one hand and a mechanical design that is in accordance with the biological principles of energy efficiency and light weight construction on the other.

The main focus of this thesis is to investigate the possibility of introducing biological principles into a control concept that is able to fulfill the above demands while minimizing design complexity. Therefore, the system principle is a divide and conquer approach that features multiple abstraction levels. This methodology allows for the complex overall system behavior to arise from the collaboration of fairly specialized and simple basic 'building blocks'. Within the architecture each level is coordinating the one below and being controlled by the one above it. This is not only implied by the biological archetypes (as presented in [Dillmann 07]) but at the same time solves the dilemma of satisfying the very different needs of the respective layers invoke concerning computational power and permissible latency.

Hence, low level reflexes that are closely coupled with the hardware and require low computational power due to their predominantly reactive character can be realized on an embedded micro-controller while the more deliberative components on higher levels in the hierarchy can be run on a physically separated desktop PC. In order to provide a homogeneous communication interface for the diverse hardware, a common bus system is required. Since the latency critical aspects of the system are located in the direct proximity of the respective actuators, the requirements concerning the real-time abilities of the bus system connecting those distributed controllers to the central PC component can be assumed as negligible in the design process. Therefore, the resulting system satisfies the need implied by the task while keeping the costs low due to usage of standardized components.

1.4 Overview

In chapter 2 the state-of-the-art approaches concerning the focus of this thesis are presented. At first various well established methods for compliant actuator control are presented. After that, hardware based approaches aiming in the same direction are illustrated. In the subsequent section approaches that adapted principles of passive dynamic machines for actuated applications are investigated. The chapter closes with a discussion of different distributed control methodologies and their biological motivation.

Chapter 3 focuses on the design of a cascaded time discrete controller. Initially a time continuous controller suited for the task is derived and its properties are investigated. After that, the possibility of transferring features of this specific controller type into the time discrete domain are discussed and the respective realization is presented. The last section of this chapter is committed to the result evaluation of the simulated time discrete controller in order to validate the achieved behavior in the context of the application.

In chapter 4 the detailed structure of the behavior based high level control architecture is deduced and the performance of the implementation is assessed using the data gathered during the simulation process. In the concluding section additional features that may contribute to enhancing the system performance for the realization are introduced and discussed.

Chapter 5 is intended to present the implementation of the entire concept on the real hardware. Afterwards the performance is evaluated in actual use are summarized. A conclusion and an outlook on the future work are given is chapter 6.

2. Related Work

The intent of the first two sections of this chapter is to introduce the state-of-the-art concerning hardware and control based approaches to realize compliant actuation. The third section presents the idea of biped locomotion based on passive dynamics. The last section is dedicated to the principles of distributed control including approaches based on this paradigm.

Compliant actuation is to be considered the key element in locomotion approaches featuring natural motions, good energy efficiency, and robustness. This is because it offers many properties that a biological actuator, i.e. the muscle, possesses. On the other hand compliance can help to reduce the potential damage to objects or persons in the working area.

None of the actuators developed so far can live up to their biological archetypes, whose power to weight ratio and dynamical properties have never been met. Nevertheless they represent means to achieve a system behavior that is roughly comparable to that of a natural system. A very central element is the spring-damper characteristics that is typical in biological systems. It allows for actuators with inherent robustness to disturbance and the ability of energy storage. Therefore self-stabilizing cyclical motions are possible that would otherwise result in a debacle. The source of these mutually intensifying disturbances is the interaction of basic motion pattern properties (e.g. frequency), external disturbances, and the systems inherent dynamics. If left unattended this will lead to undesired and unpredictable resonance effects that inhibit the realization of the desired motion.

Since power consumption is a very critical point in respect to the degree of obtainable system autonomy, approaches based on passive dynamics are a quite interesting solution for this problem. The idea behind this is to incorporate the natural system dynamics into the design and control process. Thus, no energy has to be wasted by trying to counteract the natural movements developing during repetitive motions. Actuation can rather be limited to specific points in time. The rest is taken care of by the basic laws of physics.

Distributed control is a very common feature in biological organisms. This is because it allows the control system to be very robust and scalable. The former is achieved

by making use of the locality principle. This means that the first instance of control is located in the direct proximity of the sensor and actuator. This results in a very fast control system. This is a key aspect when stability and safety are concerned.

2.1 Elastic Actuation

The following section presents several approaches to retrieve a mechanical system with tunable dynamical properties. This listing is not at all exhaustive but it gives a rather good overview of the recent developments in this field of research. There are three basic categories of actuators with the described properties. The first group is formed by actuators that employ additional controllable elements like e.g. servo motors. This allows for independently setting the position and the respective stiffness. The second category consists of actuation systems that include a series elastic element. Instead of a stiff connection between the motor and the load, a spring can be used to achieve superior compliance properties. The last group of actuators is inspired by their biological counterparts. The so called fluidic muscles consists of a special kind of mesh that forms the outer shell. Inflating this tube-like structure causes the muscle to contract. Since a single muscle can only pull by contracting but is not capable of pushing, an antagonistic setup is required for this kind of approach.

2.1.1 Actuation Allowing for Alterable Impedance

As described above, the common feature of this kind of actuators is the ability to independently control the position and the stiffness at any given time. Since there exist way too many examples of this type to discuss every single one, three very typical candidates were selected to be investigated closer. These are MACCEPA developed at University of Brussels (Belgium), VIA designed by researchers at the University of Pisa (Italy), and the AMCS approach invented at Carnegie Mellon University (Pittsburgh, USA).

The acronym MACCEPA stands for Mechanically Adjustable Compliance and Controllable Equilibriums Position Actuator and was developed by Van Ham et al.. The concept behind the actuator is depicted in figure 2.1 and exemplified in [Ham 06].

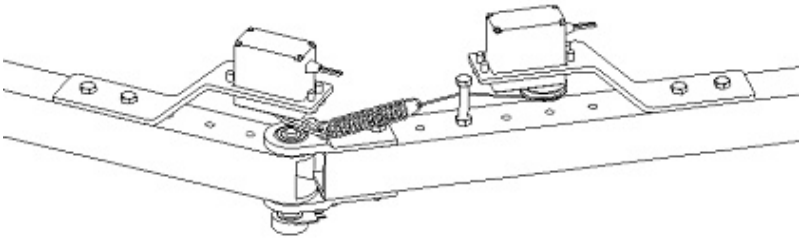


Figure 2.1: Drawing of the MACCEPA concept take from the university's website

As one can see there is one segment on either side of a common pivot point that serves as the rotation axis. A servo motor is used to set the angle between the two. A parallel spring is mounted on a cable that is suspended by a fixed point on the left segment and a lever arm extension of the right body. A second servo motor is

positioned on the other end of the cable to allow for controlling the pretension of the spring. When the equilibrium position is reached the spring no longer applies a momentum. Thus, compliance and equilibrium position can be controlled independently of each other. The influence of pretension is quasi linear. Therefore, the spring constant has a linear effect on the motion performed by the lose end of the actuator setup. Besides the common drawbacks of this kind of approaches that will be discussed in section 2.1.4, this specific setup suffers from oscillations that occur in the vicinity of the equilibrium point as soon as a disturbance or external momentum is applied to the actuator. The natural frequency of this oscillation is dependent on the compliance. This drawback is quite grave and thus, this actuator cannot be considered to be used in the jumping leg.

The second candidate is VIS which stands for Variable Impedance Actuation. A detailed description can be found in [Tonietti 05] and is depicted in figure 2.2.

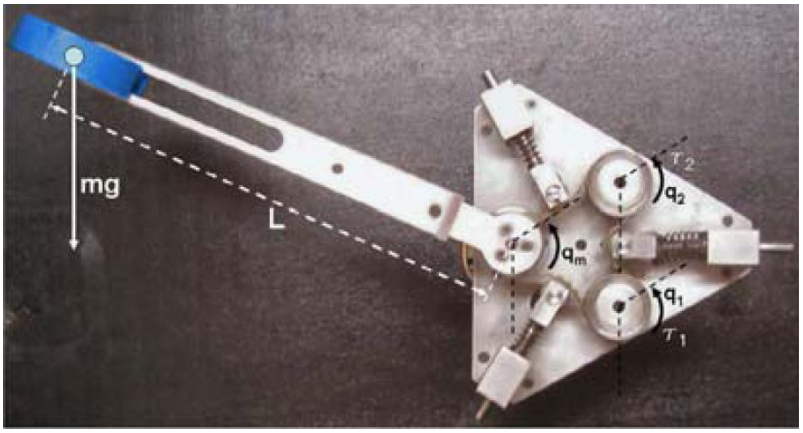


Figure 2.2: Variable Impedance Actuation (VIA) setup taken from the publication

The setup features two antagonistic pulleys that are powered by position controlled DC motors and connected via a transmission belt. Between the pulleys with the actuators the actual output axis is located. The working principle is as follows: Concordant angular variation causes simple displacement, while opposite movement causes the stiffness to change. Hence, one has to alter impedance on the fly by moving one of the pulleys. In order to ensure a sufficient tension on the belt, additional springs are employed that have no direct effect on the output. This mechanically quite complicated assembly is pretty large (approx. the size of a compact disc), heavy and control is definitely non-trivial.

The last actuator in this section is called AMASC. This is short for Actuator with Mechanically Adjustable Series Compliance. The basic idea is similar to the one of the last actuator. The details, however, are quite different as can be seen in [Hurst 04]. The intention was to develop a compliant joint with tunable natural dynamics. For this purpose two powered pulleys were connected using springs. In order to be able to alter the joint's stiffness, the springs need to possess non-linear characteristics. This is because the first actuator incorporated in the design concept is used to control the spring pretension that sets the system's stiffness while the second one controls the resting position of the device. A photograph of an AMASC device can be found in figure 2.3.

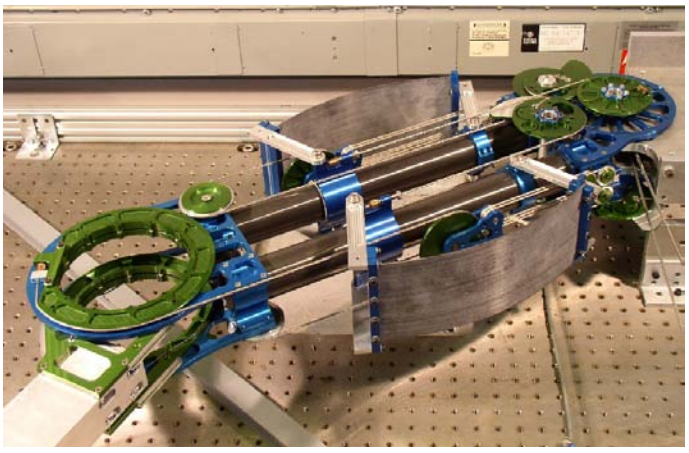


Figure 2.3: AMASC

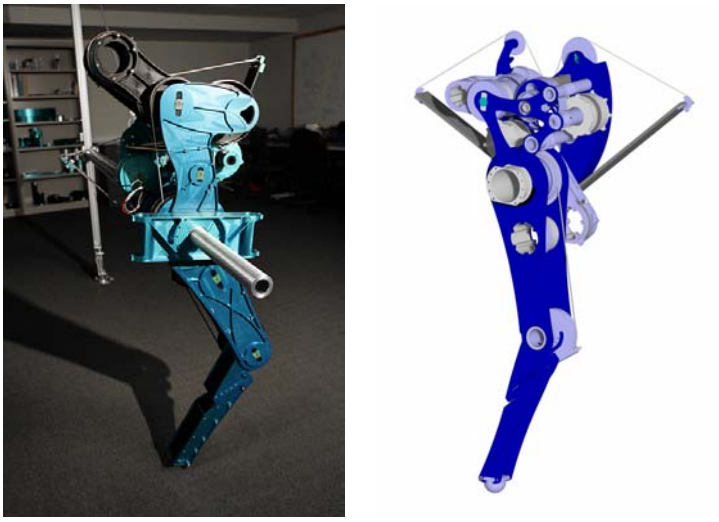
Once more the drawbacks associated with this specific actuator make it not suited for the intended kind of use in this project. Besides its mechanical complexity which makes it hard to control, it suffers from multiple other disadvantages. It can store only a very small amount of energy because only one of the two springs can actually be employed to preserve the system inherent activity. The reason for this is that the sole purpose of the second spring is to adjust the stiffness. Another effect that comes along with this is the increase in internal losses due to friction because of more moving parts. Another major drawback is the need for steadily activated pulley actuators if one's intention is to maintain the same position and stiffness (passive joint) in order to keep the springs under the right amount of tension. Another aspect of this is the surplus weight caused by the antagonistic actuator principle. In addition, an increase in stiffness decreases the available spring deflection and thus the ability to store energy dramatically.

An example of the application of a revised version of the AMASC joint can be found in [Hurst 08]. The redesign was indicated due to the inefficient antagonistic setup of the original AMASC actuator. The resulting new actuated joint, however, lacks the ability of mechanical stiffness adjustment in favor of higher energy storage capacity and reduced complexity.

The concept behind it is to design a biped that features mechanically adjustable series compliance (BiMASC). It is intended to mimic the behavior of a spring loaded inverted pendulum (SLIP) and thus achieve a circular transfer between kinetic and potential energy. The ability to store and deliberately set free large portions of the required energy reduces the amount that has to be fed into the system from outside (i.e. the power consumption). If the storage capacity is high enough only friction losses have to be compensated for. This would be a tremendous step towards achieving a human like energy efficiency.

The developed ECD (electric cable differential) leg based on BiMASC is approx. 1 m long (full extension) and has a mass of 38 kg. The peak torque available is 945 Nm with a knee stiffness that can be adjusted in between 512 and 585 Nm/rad.

A monopod version named THUMPER (figure 2.4) is used at Carnegie Mellon University to explore the role of compliance in a running gait. It reached a maximum



(a) THUMPER **(b)** CAD model of the leg

Figure 2.4: The Electric Cable Differential (ECD) leg

jumping amplitude of approx. 10 *cm*. A biped version (MABLE) is used to explore control theory for legged locomotion at the University of Michigan.

2.1.2 Series Elastic Actuation

The name series elastic actuator already reveals the main feature of the concept. Instead of a stiff connection between the motor and the load, an elastic element like e.g. a spring is used. This results in a serial connection between the two elements. The schematic concept is illustrated in figure 2.5.

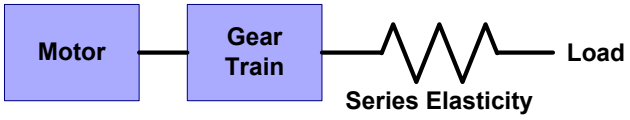


Figure 2.5: Schematic drawing depicting the series elastic principle

Details concerning the properties and application can be found in the works of Jerry Pratt et al. [Pratt 95, Pratt 02] as well as by several others like [Robinson 99] because the approach is quite popular as can be seen later on in this paragraph.

The main feature of this setup is the ability of the elastic element to low-pass filter shock loads which results in a system inherent shock tolerance that is independent of the used software control strategy. The spring used can be selected with the intention to either allow for high force bandwidth (high spring constant) or minimal friction and impedance (low constant). Hence, the spring is the most important influence of the system's overall properties and has to be selected with a specific application scenario in mind. If this is done in the correct way high force fidelity, minimum impedance, and a large dynamic range can be acquired. This will lead to a higher overall system stability and slightly increased efficiency due to energy storage capacity. An efficient force control is possible by measuring the spring deflection and applying Hook's law.

The advantages accessible through SEAs are quite clear: greater shock tolerance, lower reflected inertia, more accurate, and stable force control. They are also robust, inexpensive, offer high force to mass ratio, and can be used independently of the selected actuator type like e.g. electric or hydraulic systems. However, one has to keep in mind that SEAs are low-motion actuators that feature no stiff connection to the load. Hence, they offer only moderate force bandwidth. This means that the actor is not capable of generating a high amplitude of force at a high frequency.

As stated above, there exists a multitude of applications for this class of actuators. The most prominent one is the SPRING FLAMINGO developed at the MIT leg laboratory in the 1990s. The setup of this machine will be discussed in detail in section 2.3. Another quite interesting one can be found in [Curran 08]. A leg making use of SEA (figure 2.6) is used for a jump that is determined by a genetic algorithm (GA).

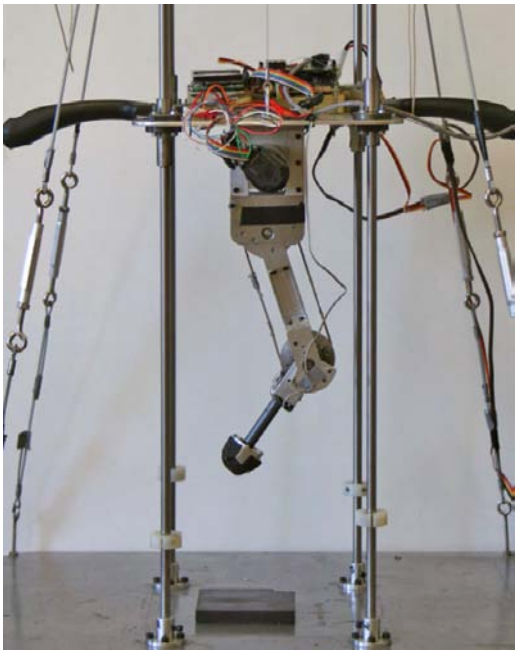


Figure 2.6: Leg with series elastic actuation proposed by Curran et al.

Besides the jump trajectory itself, the algorithm is also used to determine the best suited actuation hardware and the kinematic setup as well. Using a scalar fitness function, with 30 trials per optimization problem and 200 generations with 150 individuals per trial, the most efficient overall setup was computed. As it turns out, a counter-movement jump with elastic actuation at both joints was found to be the most promising by the computer. This seems pretty straightforward because this way the energy of the falling body can be stored in the elastic elements as it transitions from its straight starting position to the point of inflection. During the attempt to push off, this surplus energy can be employed to increase the lift-off speed and therefore the achievable peak jump height.

2.1.3 Fluidic Muscles

The design of a fluidic muscle is as simple as brilliant. It basically consists of a mesh structure incorporated into an airproof and flexible mantle. If the muscle is inflated,

the mesh causes a constriction to occur. A concept illustration of the structure can be found in figure 2.7.

Figure 2.7: Schematics of a fluidic muscle (Source: Festo)

The way the actuator works is pretty straight forward. There exists a correlation between force, length and pressure. A typical data sheet for a muscle developed by the company Festo is presented in figure 2.8.

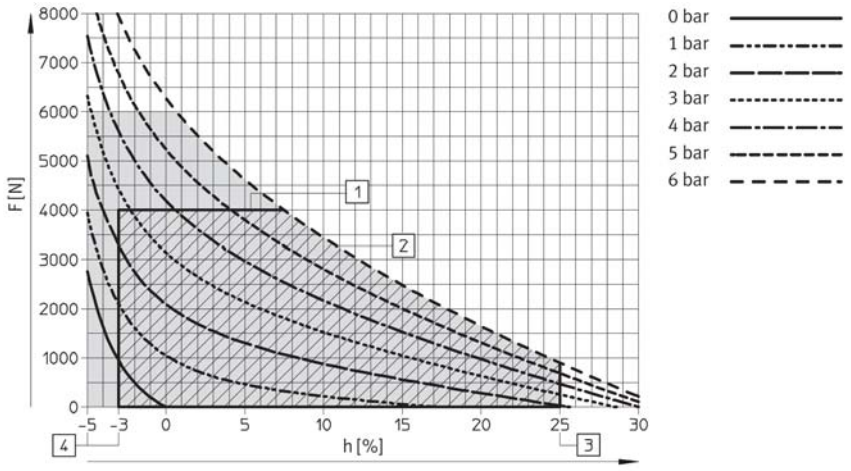


Figure 2.8: Working parameters (Source: Festo)

It basically underlines the fact that a given point (i.e. constriction) can be reached using multiple force sets as long as it is within the physical limitations of the actuator. Due to its design, a single muscle can only constrict i.e. pull but it is unable to expand beyond its initial size. Hence, an antagonistic setup is required to be able to control the position and stiffness. The former can be controlled via the equilibrium point. This is the joint position achieved when momentums of both muscles equal each other out. At the same time the stiffness is given by the force applied by either actuator. A mutual high force will result in high stiffness while lower momenta result in a more compliant characteristic.

The advantage of this actuation method is the system inherent compliance and good power to weight ratio. The trade-offs on the other side are quite severe: Besides the antagonistic principle, the strong non-linear system dynamics and control characteristics of the required hardware (like e.g. valves) contribute to the fact that this class of actuators is very hard to control. A promising control methodology is presented in [Kerscher 05]. Here the system is modeled as a spring with a parallel damper

to reflect the dynamic properties. Caused by the non-linear characteristics, a linearization around a fixed working point has to be performed. This works fine if the operation area of the muscles can be kept in the close proximity of this point but causes rather significant deviations if that demand cannot be fulfilled. Another application can be found in the already mentioned robot LUCY in [Vanderborght 06]. In order to be able to efficiently control both position and stiffness of the actuator, a PID pressure control is used as the innermost loop below a torque and trajectory controller. Furthermore, power consumption was reduced by introducing an algorithm that is capable of matching the fitting actuator compliance to the natural compliance of the trajectory.

2.1.4 Summary of Hardware-Based Compliance Approaches

In this section an overview over actuating systems with inherent mechanical compliance was given. The central arguments in favor or against the respective representatives are summarized in table 2.1.

Name	Advantages	Drawbacks
AMASC	linear behavior high bandwidth weight & energy consumption	tuning dynamics non trivial hysteresis
MACCEPA	dyn. alterable compliance linear for wide angle range cheap	oscillation around equilibrium low momentum weight
SEA	compliance energy storage	low bandwidth fixed stiffness
Fluidic Muscle	inherent compliance no gear required low bandwidth	antagonist required hard to control cost restrains autonomy

Table 2.1: Overview of compliant actuation based on mechanical solutions

The first two actuators belong to the category of actuators that offer mechanically alterable stiffness settings. Besides the already mentioned complexity and respective individual weaknesses they all share some common drawbacks: Energy has to be 'wasted' on setting or maintaining a stiffness setup which the motion task cannot directly benefit from. Along with this comes the extra weight of the second actuator and a complex setup. Therefore, they are not at all suited for the specified task since both energy consumption and weight are two very critical factors in highly dynamical applications that influence each other in a kind of vicious circle. Higher weight requires more actuating force due to the surplus inertia that has to be overcome. Hence, the actuator has to be exchanged for a bigger model which again is heavier and so on.

The concept of series elastic actuation seems very promising although the actuator itself still lacks one very important feature: sufficient energy storage capabilities. Since the elastic element is predominantly used to serve as a low-pass filter for shocks like e.g. the reflected inertia in an impact situation, it has to be dimensioned accordingly. This however mostly contradicts the use as a device for energy storage because this would require different spring parameters. Hence, the task of the actuation concept for this project has to achieve a comparable result in respect to shock absorbance with an improved energy storage potential.

The last class of suggested actuators is based on the principle of McKibben muscles and although it offers many of the required properties it is not suited for the application. This is mainly because of the the limited bandwidth and the high overall weight considering all the peripheral hardware that is required like e.g. vales. The fromer is of special importance since this application requires fast transitions between system states which implies a fast actuating element. Unfortunately this cannot be assumed for the actuators at hand. Another aspect is the very complex and specific control as well as the physical actuator limitations in respect to the correlation of muscle constriction and force. This means that only a limited number (working area) of different force sets can be achieved for a given position. The last major disadvantage is the reduced degree of autonomy since the actuator is dependent on a steady supply of compressed air to inflate the muscles. Therefore, this kind of actuators does not seem to be an option for a jumping robot either.

As can be seen, a solution to achieve compliance that entirely focuses on hardware is not feasible here. Thus, one has to investigate the potential of actuator control methodologies that allow for incorporating compliance. According to [Pfeiffer 07] the question whether to use hard- or software approaches comes down to the question of high actuator bandwidth vs. complex mechanically designed systems with tuned natural dynamics. The former is very flexible but it is hard to find a suited actuator while the latter is inflexible and requires extensive considerations of dynamics prior to construction but allows for simple control algorithms.

2.2 Joint Control Methodologies

In the beginning of robotics the idea was to make the joints as stiff as possible in order to reliably use e.g. robot arms to manipulate objects during a manufacturing process. This was mainly because biological principles were not considered to a large degree since there was no need for compliance. Another aspect of this was that approaches for such an undertaking were not available at that time. This is because control theory, design, and simulation tools were not as well developed as they are today. When biological principles were adapted into the field of robotics, a change of paradigms occurred. This was the case since one came to understand the advantages of selective compliance. Among these are enhanced energy efficiency, safety aspects in collision situations with humans or objects, and less wear and tear.

During the last couple of years, several approaches in the field of control theory were established to achieve such behavior. Three major groups of those approaches will be presented in this section. First of all, old fashioned stiff trajectory control is presented to serve as a basis for comparison. In the adjacent section the class of

impedance control methods is investigated. The last two sections are dedicated to the computed torque method and virtual model control.

2.2.1 Trajectory Control

This control methodology was developed with process automation tasks in modern factories in mind. For this purpose e.g. robotic arms had to be repeatedly moved along an initially taught-in trajectory. The main concern was to be able to follow the desired trajectory as closely as possible in order to ensure proper results of e.g. automated welding or painting. Thus its intent is to imitate a predefined motion without further side conditions.

Since the simple copying of movement patterns has shown not to be suitable for applications in the field of walking robots, it can be complemented with the zero moment point (ZMP) approach in order to improve the static stability of the machine. A detailed description of the ZMP method can be found in [Kajita 07, chapter 3]. The ZMP itself can be defined as the point where all impulses equal each other out and thus the resulting movement is zero. This means the ZMP is the contact point between the foot and surface where impulse from ground and the impulse caused by the robot motion entirely equal each other out. The approach also includes a measure for stability. This is very important in order to be able to predict and counteract unsafe situations before they actually happen. Therefore the so called stability region is introduced. It represents the area where the robot maintains steady contact to ground. The robot can be considered to be stable as long as the ZMP is located within this region. Furthermore the distance between the ZMP and the border of the stability region can be employed as measure for stability.

Computation of the ZMP requires solid model knowledge such as the robot pose, absolute joint velocities and angular velocities. The task of generating a proper model for the robot is very crucial since all modeling errors as well as noisy sensor data will propagate into a deviation of the ZMP from its actual location. More than that, the ZMP approach fails if either slip occurs, uneven ground is used, or a collision of the arms, torso etc. with objects in the environment occurs. This is because events like those above will generate impulses that are not considered within the model and thus are not compensated for. This will destabilize the robot or in the worst case cause it to tilt over.

Hence, it can be stated that this relatively straightforward approach is only suited if the very narrow requirements can be met. This, however, is only possible in very sparse and structured laboratory conditions. Application in an everyday scenario is therefore impossible.

The first bipedal walking robot using this method to gain major public interest was the already presented ASIMO. Besides this one however, there exists a variety of other examples. One of them is the monopod KENKEN introduced in [Hyon 02] that is intended to be used in a running robot. A photograph of an earlier project stage is presented in figure 2.9.

It is inspired by a dog leg. Its mass is 13.3 kg and its height in full extension is 52 cm above ground. It consists of two hydraulic actuators with one parallel spring modeling the Archille's tendon. An empirical controller featuring separate

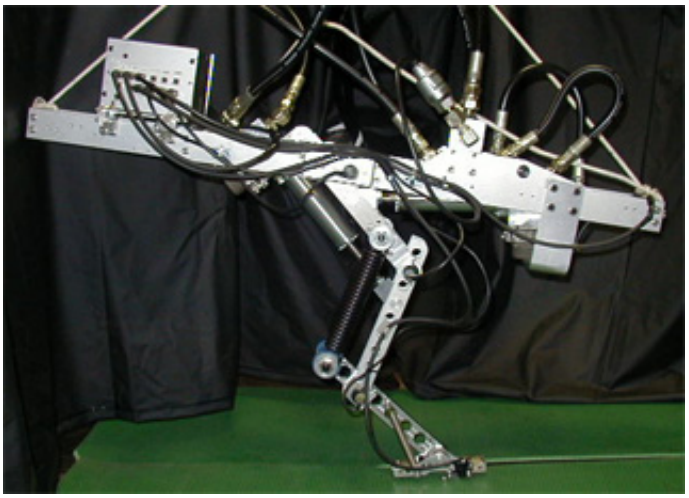


Figure 2.9: KENKEN: A monopod intended for running

trajectories (that were generated using simulation data) is employed for each phase. A FSM was utilized as a high level coordinating function that is capable of managing the phase transitions. The main purpose of the spring is to compensate shocks that the stiff actuation is not able to deal with. With this approach, a very common problem of stiff control methods occurs: At higher speeds the reflected inertia and not properly modeled dynamical features cause the robot to become unstable.

2.2.2 Impedance Control Methods

Impedance control forms a wide class of compliant control approaches that allow for simultaneous control of position and force. In comparison to the aforementioned method this puts one in the position not only to control the Cartesian location of e.g. a robot manipulator but also allows to set a reference input for the force that is to be applied to get there. This offers a major advantage, since by this means one is able to perform more delicate tasks with more sensitivity instead of trying to reach the desired trajectory with brute force. Besides the potential to open up new fields of application for a robot, this approach also allows the user to incorporate safety aspects in a manipulator program. Therefore, robots can be applied in environments where collisions etc. can occur.

As pointed out in several works that focus on bipedal walking ([Lim 01, Park 01]), the properties of these approaches are the key to achieve human like motions. This is because humans make heavy use of the spring-damper properties of muscles to reach energy efficient and robust locomotion. Therefore the idea behind impedance control is to establish a mass-damper-spring relation between the Cartesian position Δx and Cartesian force f. Using the damper constant d_k and spring constant K_k, one is able to denote this like presented in equation 2.1.

$$f = M\Delta\ddot{x} + D_k\Delta\dot{x} + K_k\Delta x \tag{2.1}$$

with M being the inertia matrix. This equation has to be solved for the given reference inputs in order to realize the desired motion. Three methods summarized

in [Albu-Schaeffer 02], that try to solve this exact problem are presented in the next paragraphs: Admittance-, impedance-, and stiffness control.

2.2.2.1 Admittance Control

This method is amongst the most commonly used since only the existence of a position interface is required to realize the control task as formulated in equation 2.2.

$$x_d(s) = x_0(s) - \frac{\Delta f(s)}{K_k + D_k \cdot s} \rightarrow \tau_m = P_R\{K^{-1}\{x_d\}\} \tag{2.2}$$

For this purpose the force vector is measured in order to generate the desired Cartesian position x_d. Applying the inverse kinematics K^{-1} one is able to convert the resulting position deviation into the desired joint positions. Once this is done, the joint position controller P_R is utilized to determine the required torques τ_m.

This approach results in rather good position accuracy. Besides that, it is capable of compensating the respective joint friction using a high gain position controller. For low desired stiffness and damping however, stability problems might occur since the controller bandwidth will become about equal to the joint bandwidth.

2.2.2.2 Impedance Control

Impedance control makes direct use of equation 2.1. The Cartesian position can be computed based on the joint position q_2 using direct kinematics. Now the transposed Jacobian can be employed to transform Cartesian force into joint torques. Thus, a torque controller T_R can be used to set the commanded momentum with the motor.

$$f = K_k \cdot \Delta x + D_k \cdot \Delta \dot{x} \rightarrow \tau_m = T_R\{J^T(q_2)\} \tag{2.3}$$

In general, good results can be achieved using this controller technique. This is because it is able to overcome the disturbance caused by joint friction. It is ideally suited for low stiffness and damping since torque controller bandwidth is optimally exploited. Stability problems might occur for high desired stiffness. Therefore, it can be stated that this approach is in many aspects contrary to admittance control.

2.2.2.3 Stiffness Control

The idea behind stiffness control is to convert the desired Cartesian stiffness and damping into the corresponding joint stiffness and damping matrices K_j, D_j. The generated desired joint torque can be controlled using joint impedance controller S_R.

$$\tau_m = S_R\{K^{-1}(x_d), K_j, D_j\} \tag{2.4}$$

Since the computation of the two matrices includes the Cartesian displacement, the interval between two computations is critical in order to keep the resulting error below a reasonable threshold. For high displacements the error is rather large anyway.

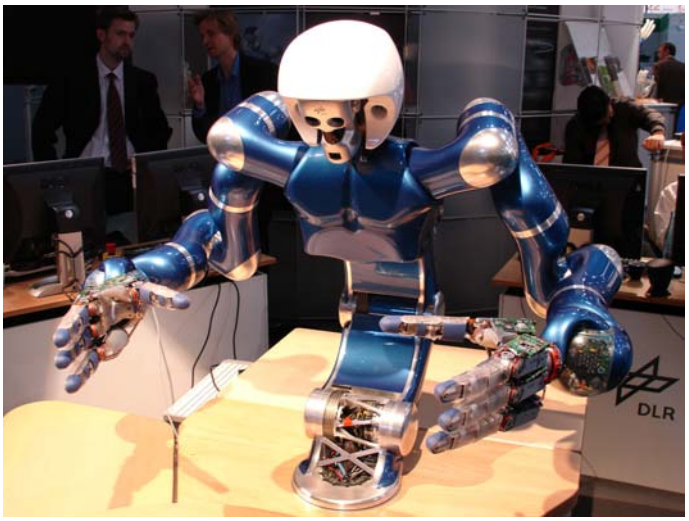


Figure 2.10: Light weight robot JUSTIN (source: www.dlr.de)

An application of impedance control was realized at the German research center for aviation and astronautics (DLR). The robot JUSTIN with two light weight arms is depicted in figure 2.10.

The control concept of the arms features an impedance controller combined with local stiffness control. Thus, the robot is able to benefit from both worlds: High positional accuracy can be reached while offering a high range of attainable stiffness. It is structured into a slower Cartesian loop responsible for computing the dynamics and inverse kinematics and a fast joint control loop. This is indicated to reduce displacement per step and therefore keep the deviation of the impedance controller low. Due to the high computational load, a distributed approach has to be taken. Thus, the slower loop is implemented centrally while the joint loop is realized locally. The controller behavior is continuously adjustable by simply setting a parameter to show more torque- or position controller-like performance. Although an admittance controller can achieve better geometric accuracy the presented approach is superior when high bandwidth is required.

2.2.3 Computed Torque Method

The last of the classic control theoretical approaches is the computed torque method as presented in [Loeffler 04]. It allows for computing the entire theoretical system dynamics using equation 2.5.

$$M\ddot{q} + W_1 \cdot \begin{pmatrix} \lambda_m T_x T_y \end{pmatrix} = h + W_F \cdot \lambda_{FR} \tag{2.5}$$

with

	M	-	mass distribution matrix of the system
	q	-	generalized coordinates
	h	-	vector combining the non dynamical terms
	T_x	-	lateral torque between supporting foot and the ground
	T_y	-	frontal torque between supporting foot and the ground
	λ_m	-	torques of remaining joints
	λ_{FR}	-	forces of supporting foot and torque around vertical axis
	W_1, W_F	-	Jacobians mapping torques and forces to generalized coordinates

Assuming a given desired trajectory defined in Cartesian space, one is able to compute the required motor torques. While doing so, the system limitations e.g. in respect to the maximum torques acquirable through the actuators, are considered. The Jacobians are then used to map the torques to generalized coordinates. Since the system limitations are already incorporated in the controller definition, stability is inherently guaranteed.

Although this method theoretically produces the optimal system performance since all dynamic effects can be considered, it has some major drawbacks: It is not suited for fast motion application due to the low overall system bandwidth. This is caused by the high computational effort as well as the latency in information propagation (sensors as well as actors).

2.2.4 Virtual Model Control

A non-traditional approach was introduced by J.E. Pratt ([Pratt 01]) in 1995 at the MIT. It is intended to make the hard task of controlling bipeds easier by reducing the design complexity. For this purpose virtual components (VC) are introduced. In order to make it more intuitive the VCs represent well-known, every day physical concepts like springs, dampers, masses, latches, bearings, and non-linear potentials. Each VC is connected to a specific point on the robot model and a target position. Thus, external forces are generated and applied to the system. In order to determine the joint momentum that is needed to compensate for the external force, Jacobians are used in an elegant way in the kinematic chain. Once all intra segment forces are determined, the momentum propagation into the adjoining segments is computed. For this purpose, the first frame is considered the action frame, i.e. the segment that is intended to perform motion, while the adjoining one serves as the reaction frame.

It is to be pointed out that neither inverse kinematics nor a dynamic model of the robot is needed. This is quite remarkable since it does not only allow for unfolding of the natural system dynamics and natural movements but also requires little computational performance. Since there is no need to perform linearization around a working point, modeling errors can be entirely avoided. Like the biological role model the process requires only very sparse sensor information.

An application of this method in the field of walking robots can be found in [Kerscher 07]. The key idea is to initially set up a static model and later on extend it to a complexity level that is able of handling the dynamic motions by adding further VCs. Initially one VC per leg is used to compensate for gravity influence. In order to control the additional DOF more VCs have to be added. For this example 23 VCs are needed for mastering the walking task. The relatively low number is made possible by continuous re-parameterization, activation, and de-activation through a high level FSM as well as the fact that fluid muscles are employed. Due to their inherent compliant nature there is no need to explicitly model it in the controller. However, this causes an additional overhead for stiff actuators like DC motors when compliance is required.

2.2.5 Resume on Control Methods

In this section four classes of control algorithms were introduced. Their respective advantages and weaknesses are summarized in table 2.2.

Method	Pro	Con
Trajectory Control	intuitive design versatile	poor energy efficiency dyn. model required high computational load
Computed Torque	theor. optimal performance	poor results for fast motion
Virtual Model Control	simple and unorthodox natural motion natural dyn. are considered	physical intuition needed design not trivial
Impedance Control	high bandwidth	poor geometric accuracy inverse kinematics needed

Table 2.2: Overview of control methods intended for compliant control

The old-fashioned and naive approach of trajectory control has not proven to be suited for the task at hand. This is for multiple reasons: First of all, a very precise model of the robot and its environment is needed in order to keep the deviation level low. The direct result of this is the need for precise sensors and high computational power. Undesired side effects for a jumping task of this kind are that one is only capable of leaving ground at the cost of enormous energy expense and relatively unpredictable impact dynamics. Therefore, it is not suited to be used in this context.

Impedance control on the other hand is capable of achieving compliant behavior. This approach, however, still makes use of inverse kinematics which counteracts the ability to employ natural system dynamics. Although it is well suited for highly dynamic applications the overall geometric accuracy is low. The properties of the computed torque method are directly opposed to that. It is capable of producing excellent results in theory but is unable to work properly with fast movements.

The approach taken for the virtual model method is entirely different to the ones above since it does not employ well established tools of control theory that are common to the others. Instead it requires physical intuition to allow for an easier design process. Here the dimensioning of the utilized components as well as the fixture point are quite critical. The unique feature about this approach is that the controller design can be extended later on in order to be able to cope with new situations instead of having to repeat the whole design process. The main drawback, however, is that the modeling of compliance for non compliant actuators is very time-consuming. This is not necessarily the fault of the approach since it was developed to be used with SEAs. It did prove its potential while being used in the robot SPRING FLAMINGO.

In conclusion, one can state that an impedance controller-like behavior is desirable for this project. The need for inverse kinematics, however, has to be overcome in order to allow for natural dynamics to unfold.

2.3 Approaches Inspired by Passive Dynamics

The idea of making use of passive dynamics in the context of bipedal locomotion is based on the works of Tad McGeer [McGeer 90, McGeer 93]. He investigated the properties of a system's natural, unforced dynamics in the context of walking mechanics. The interpretation of Collins et al. [Collins 01] of McGeer's approach is that he designed walking the way the Wright brothers designed airplanes. Thus, at first one has to understand the inherent dynamics of the passive systems. Once this is mastered and understood one can go ahead and add as little actuation as possible. The actuation is needed in order to be able to create a more versatile machine.

Entirely passive walkers only work on perfectly even and sloping ramps, since they are dependent on the transformation of potential into kinetic energy. This is because they are lacking proprietary sources of actuation. It is crucial to limit the amount of actuation to a degree that is not capable of overpowering the inherent dynamics like one can find it in many trajectory control based approaches. The key is to add actuation only where it is required in order to stabilize the repetitive motion and compensate for loss. For the rest of the movement pattern one has to allow the system to take care of itself.

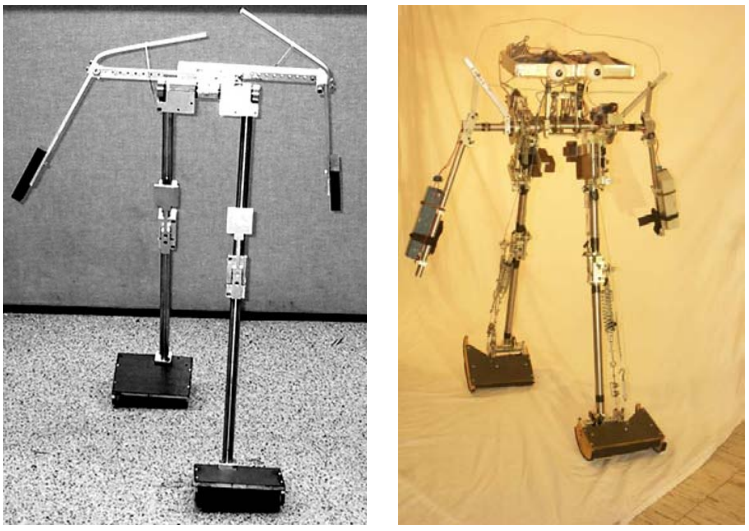
One of the main aspects of this methodology is the very high energy efficiency. This can be even enhanced when compliance is considered. This is because the main source of energy loss in the system besides joint friction is the ground impact. This loss can be reduced by storing the energy in elastic elements in the joints rather than having to deal with it in form of undesired reflected inertia. The advantage of combining compliance with natural dynamics besides reduced energy consumption is increased versatility and better adjustment to disturbance which again allows for reducing the control overhead as stated in [Anderson 05].

The application of these ideas (amongst other features) is pursued by Martijn Wisse in his dissertation [Wisse 04] at TU Delft. He and his group developed several walking machines like FLAME (figure 1.3(b)) and DENISE. The result of another attempt based on passive dynamics can be found in [Collins 05a]. The design of the Cornell biped (presented in figure 2.11(b)) based on an initial entirely passive walker (figure 2.11(a)) is described.

As one can see, the basic features like the kinematic setup of the passive walker (left) was adapted and expended by actuators and elastic elements to form a more versatile walking machine (right).

In order to be able to discuss the effectiveness of such an approach in a reproducible way, the power consumption has to be numeralized. This was performed by Collins et al. at Cornell University. As presented in [Collins 05b], regular walking machines are compared to those based on a passive dynamic approach and then put in relation to a human being. The overview can be found in table 2.3.

In this overview, the specific cost of transport is broken down into mechanical c_{mt} and electrical costs c_{et}. In order to provide a fair base of comparison, the values are normalized in respect to weight and traveled distance. As one can clearly see, passive dynamics based approaches have a significantly lower energy consumption than a classical trajectory based robot like e.g. ASIMO. As the Cornell biped demonstrates, they can achieve almost human-like effectiveness.



(a) Cornell passive walker **(b)** The Cornell biped

Figure 2.11: The design of the Cornell biped (right) was based on its predecessor, a passive walker (left).

Test Subject	Mechanical Cost c_{mt}	Electrical Cost c_{et}
ASIMO	1.6	3.2
DENISE	0.08	5.3
SPRING FLAMINGO	0.07	2.8
Cornell Biped	0.055	0.2
Human	0.05	0.2
Passive Walker	0.04	–

Table 2.3: Overview of locomotion cost as presented by Collins et al.

The application of passive dynamics based approaches is not dependent on the kind of actuator that is used. However, the utilization of an actuator that offers compliant properties by itself helps to reduce the complexity. As demonstrated in [Vanderborght 07], pneumatic artificial muscles are ideally suited for this because of the inherent passive compliance.

The advantages in employing a passive dynamic based approach can be summarized to four major points: First of all, it helps to reduce the energy consumption sufficiently during repetitive motion sequences like e.g. walking or jumping. Along with this, the task of achieving natural, human-like motions can be simultaneously solved since nature's design principles work the same way. Therefore, the resulting motions are similar if the dynamic models are roughly equal. The third positive side effect can unfold if these principles are combined with compliant joints. This combination allows for very robust and easy to control movements due to predictable dynamics and inherent disturbance compensation. The last positive aspect is amongst the most important ones for the jumping task. Due to the dogma of limiting the actuation to a minimum an enormous potential of weight saving arises. This is because

actuators have the tendency to become significantly heavier with increasing peak output level. Hence, a better peak jump height can be achieved with less energy expenditure.

Although the advantages outbalance the disadvantages they should not remain unmentioned here. The major drawback associated with a passive dynamics based approach is a significantly more complex design process. The kinematic setup with its dynamic properties has to be considered from the very beginning of the development process on. All other components like the actuators and the control architecture have to be designed around that in order to function properly as one unit.

2.4 Distributed vs Centralized Control Paradigms

The last aspect that this section is going to focus on is the introduction of distributed principles into the control architecture. As the developments of the last years have shown, the amount of DOF is steadily increasing. This pays tribute to the goal of allowing walking machines to become more versatile and comply with more elaborate tasks that are performed. Along with this goes an increase in computational load and a problem concerning the wiring. The most obvious solution is to tend to biological principles once more as shown in [Espenschied 96]. The author presents the concept of distributed control with local reflexes in common insects like the migratory locust (*Locusta migratoria*) depicted in figure 2.12(a). In accordance to the biological archetype, the principles are transferred onto a six legged insect-like robot presented in figure 2.12(b).

(a) Migratory locust (source: Wikipedia) **(b)** Hexapod developed by Quinn et al.

Figure 2.12: Hexapod robot with distributed control architecture and its biological model, the migratory locust

The concept of locality is a key aspect of the overall approach. This is because in nature and in engineering alike, close local coupling of an actuator and its control system is equivalent to a fast adaptation to external distortion. This is of special interest in the context of closed loop control. Here latency is the most crucial factor when it comes to stability since a delay in sensor information or the actuating element might cause the controller to get unstable. More than that, distribution makes the system more robust to component failure. If e.g. the central coordination unit is malfunctioning, all distributed functionality will still be operational. This way the system can be brought to a safe halting position.

Another positive aspect apart from safety and latency aspects is the increasing demand for computational power on an autonomous machine with limited space, energy and weight. In order to be able to cope with modern bipedal walking robots that make use of a centralized control architecture, the processing unit has to be equipped with surplus performance to compensate for the latency caused by the physical distance to the process. This can be avoided by using distributed components. This way each processing unit can be tailored to the specific needs of the local process. This helps to reduce the power consumption.

The inter-unit communication can be realized using a common bus system like e.g. FLEXRAY, CAN, I2C, or ETHERNET. This also solves the wiring problem that arises with modern machines. Instead of having to use shielded cables that have to run through the entire body construction, only very short distances have to be bridged. This solves the routing problem and reduces the weight since the locally preprocessed data can be exchanged via one central bus cord.

The downside of this again is the fact that a distributed architecture is harder to design than a centralized one. This is because in the former case concurrency aspects have to be considered that can be neglected with the traditional methodology.

The concept of distributed control in robots has been applied in various machines. The aspect of latency in a close-loop control system was investigated by Rizzi et al. [Rizzi 92] on a 3 DOF robotic juggler. The results underlined the demand for low latency in order to ensure stable controller behavior. A distributed architecture was also used in the robots of the HRP series designed by research groups in Japan. The concept behind HRP-2 is presented in [Matsui 05]. Here the trajectory is controlled with a loop time of 5 ms by a central processing unit while the motor loop time was set to just 1 ms. The computational power assessment resulted in an estimated need of about the equivalent of a Pentium 4 at 3 GHz. This demand was met with a highly scalable distributed real-time multi-threaded network. It makes use of the so called RESPONSIVE LINK system. This offers two channels that are used to independently transmit data and event messages. Furthermore, it supports prioritization, error detection and a bus clock of 800 MHz.

The second application of this methodology is presented on the successor platform: HRP-3P. It was designed and implemented by Kanehiro et al. [Kanehiro 06]. The processing system is comprised of a master CPU board and several I/O slave boards. RT-Ethernet is used as a common data bus for component communication.

3. Compliant Low-Level Actuator Control

In the following sections the low-level actuator controller scheme is introduced. At first, the basic question whether to use classical linear or non-linear control is discussed. The subsequent section presents the multi-loop control structure selected for this application. The control laws are derived for the time continuous domain and then transferred into the time discrete domain of the target platform. For this purpose the plant as well as the power supply and the controller is simulated in MatLab Simulink to allow for closer investigation of their respective properties. Once this is dealt with the overall controller performance is assessed.

3.1 Linear vs Non-Linear Control of DC Motors

The control of DC motors has been and still is a very popular topic and heavily discussed problem in the field of applied control theory. Two of the multiple reasons for this are that first of all it is definitely a non-trivial problem and the second reason is that the number of different actuators and application types is nearly infinite. The basic question that has to be answered before one can proceed to the control task itself is the kind of controller design process that fits the specific needs of the application best. Thus, two concepts have to be considered and evaluated in order to find the best suited one: linear and non-linear control.

Generally speaking, non linear control is the control of non-linear plants, i.e. systems whose behavior cannot be described using linear equations. At first glance this might not seem too bad but as a consequence all those well established analysis tools like the root locus, Bode plot, pole placement and Nyquist criterion cannot be applied for this case. This is due to fact that their inherent methodologies fail for non-linear systems. Therefore, non-linear control is harder than control of the familiar LTI-systems[1]. Even though it is inconvenient for some applications, such discomforts are unavoidable due to the otherwise appearing lack of precision that is

[1] LTI = **L**inear and **T**ime **I**nvariant

required. In case of the specific application discussed in this thesis, various arguments seem to be in favor of a non-linear solution. Among these are e.g. persistent modeling uncertainties, drift of electrical components of the motor and the lack of full observability. A good overview over various methods like reduction of the dynamic order or piecewise functions as well as the energy shaping approach can be found in [Taylor 94]. However, the numerical overhead caused by such approaches in comparison to a linear system description and the therefore decreasing performance should be taken into consideration as well.

On the other hand there are several arguments that lean towards linear control structure: A non-linear control system is tailored to a specific system and cannot be transferred to a different one. This is because the unique properties of each system have to be incorporated into the design process. In this case however, a general solution suited for various types of DC motors has to be found in order to ensure the possibility of reuse in other projects based on different hardware. Another argument in favor of a linear control solution, besides the straightforwardness of the design based on well established methods, is the fact that in this specific scenario the demand for precision on the controller level can be kept moderate. The trade-off of this is a more simple, broad and robust design process. Potential errors will be compensated by the high level control architecture anyway without any additional effort. Hence, a linear control scheme based on cascaded controllers was selected because it is a powerful means of controlling a complex system while keeping the design process as well as the implementation as simple as possible. In the following sections, a time continuous control algorithm will be derived and subsequently discretized in respect to time in order to allow implementation in the already established DSP control structure. Once this is done, a simulation environment for MatLab Simulink will be provided in order to be able to test the controller under controlled lab conditions before it is finally implemented and tested on the actual hardware.

3.2 Continuous Controller Modeling

In order to derive a controller suited for the given plant to control (i.e. motor), several parameters have to be determined at first. To be in a position to provide a more structured and theory based approach to the design task, various methods suggested by other authors were analyzed. The one that proved to be best suited for this specific task can be found in [Pfaff 92, p.40-81]. It is based on a multi-loop structure presented in figure 3.1. The inner loop is responsible for controlling the current while the outer loop is designated to control the position. The structure of the two inner loops is illustrated in figure 3.2 and 3.3. The controller is completed by the outer position control loop that employs the speed controller as a means to be able to reach the desired position.

The specific type of controller setup is well established throughout literature and can also be found in e.g. [Gevatter 06][2]. This approach offers several advantages over a standard control loop:

[2][Gevatter 06] chapter 5 - Elektrische Antriebstechnik (L.Sack), P.665 ff

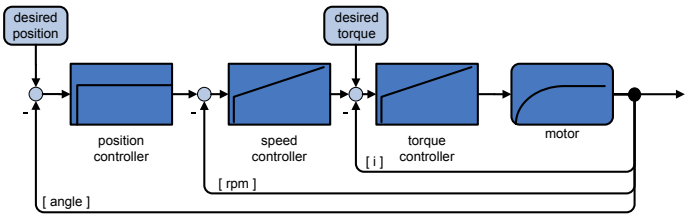


Figure 3.1: Cascaded controller structure

- The influence of any disturbance like e.g. an external torque caused by inertia of the mechanical components of the leg has only local influence. Thus, in case of the given example only the inner loop is burdened with the disturbance while the outer one does not have to deal with the effects of it.
- Setup and optimization of the controller parameters can be performed one after the other. Hence, cross-correlation of the loops can be almost entirely ignored for the implementing process.
- The command variable of the inner loop can be bounded by means of the outer loop. This allows for superior controllability and protection of the inner loop and the hardware.

3.2.1 Current Control Loop

The innermost structure of the control setup is the current control loop. As one can see in figure 3.2, it is quite simple and consists of a controller, an actuator, the motor and an optional smoothing element used to ensure that the fed back current signal is properly conditioned for the control task. For this application the most suitable kind of controller is a regular PI controller. This specific type is highly indicated in this case since a potential will occur at the motor windings acting as inductor as soon as it starts to rotate. The result of this is a current that is proportional to the motor's speed and directed against (polarity) the armature current as described in Lenz's law. This effect is modeled as a disturbance in the inner loop. In order to be able to compensate this, an integral component is required in the controller to eliminate the deviation from the reference input. The actuator is modeled as a pure delay element while the motor itself shows first-order time delay (PT1) element behavior.

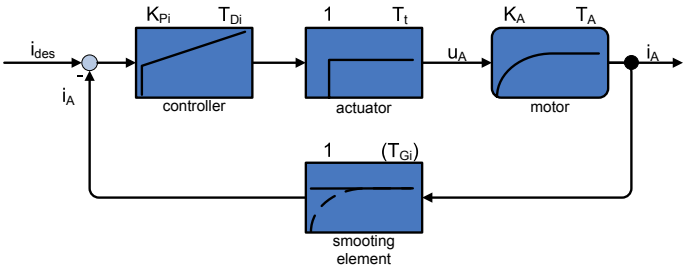


Figure 3.2: Structure of the current control loop

Now that the overall control structure is determined, the setup and parameters of the loops become an issue. Before the transfer function of the current (i.e. inner)

control loop can be found, several characteristic parameters have to be derived and estimated in order to receive the former. Since the overall task is to control a motor with a shrunk-on-disk rotor (Heinzmann SL120-2NFB), the specific parameters, as provided in the motor data sheet, are used for the deduction of the transfer function parameters. First of all the mechanical time constant hast to be found, i.e. the delay until the idle speed is reached once the motor is started. It can be calculated as follows:

$$T_{idle} = \frac{2\pi \cdot \theta \cdot n_0}{M} = \frac{2\pi \cdot 3.5 \cdot 10^{-4} \cdot 58.25}{1.3} \left[\frac{kgm^2 \cdot 1/s}{Nm}\right] \approx 95.7 \cdot 10^{-3} s \tag{3.1}$$

where $n_0 = 58.25\ s^{-1}$ is the idle speed, $M = 1.3\ N$ is the motor toque and $\theta = 3.5 \cdot 10^{-4}\ kgm^2$ is the armature's moment of inertia as specified in the motor data sheet.

In a similar manner, the time constant for the armature current (T_A) can be determined. With $L_A = 85 \cdot 10^-6\ H$ being the armature impedance and $R_A = 0.31\ \Omega$ the respective resistance, T_A can be estimates as

$$T_A = \frac{L_A}{R_A} \approx 0.274 \cdot 10^{-3} s \tag{3.2}$$

Based on the structure of the inner loop, the overall behavior of the remaining loop time constants T_t (actuator delay of the current loop) and T_{Gi} (time constant of smoothing element) can be expressed by a single one as:

$$T_\sigma = T_{Gi} + T_t \tag{3.3}$$

Since our long term task is to ensure the reproducibility of the results found for the time-discrete controller, the respective values as can be found in the real hardware will be used here. Thus, T_{Gi} can be approximated with the DSP control-cycle loop time representing the minimum time interval where a change of the input value can be detected and T_t can be assumed to be the smallest possible reaction time for a change in the armature current, i.e. the cycle-time of the PWM unit. Hence, in this specific case we receive $T_{Gi} = \frac{1}{1000}\ \frac{1}{Hz} = 10^{-3}\ s$ and $T_t = \frac{1}{39000}\ \frac{1}{Hz} \approx 0.2510^{-6}\ s$. It is obvious that the summed time constant is solely dominated by T_{Gi} because $T_{Gi} \gg T_t$ holds true. Thus, $T_\sigma \approx T_{Gi} = 10^{-3}\ s$ will be used from now on.

Based on the structure model, the transfer function of the open inner control loop can be concluded as

$$G_0(s) = k_{Pi} \cdot \frac{1 + s \cdot T_D i}{s \cdot T_D i} \frac{1}{1 + s \cdot T_\sigma} \frac{K_A}{1 + s \cdot T_A} \tag{3.4}$$

Assuming that the control response time T_{Di} for the current loop is approximately the same as the electrical armature, i.e. $T_{Di} \approx T_a$, equation 3.4 can be simplified to

$$G_0(s) = \frac{k_{Pi} K_A}{s \cdot T_A + s^2 \cdot T_\sigma T_A} \tag{3.5}$$

Thus, we receive the closed loop transfer function as

$$G(s) = \frac{1}{1 + G_0(s)} = \frac{1}{1 + s\frac{T_A}{K_{Pi}K_A} + s^2\frac{T_A T_\sigma}{K_{Pi}K_A}} \tag{3.6}$$

Under the assumptions made, the current control loop can be modeled as a second order system. This is a very common approach for the description of a physical system. Performing analysis based on the time constant T and damping factor d of such a system is a well known and mastered problem. Ideal command action of the control loop could be observed if the frequency response $|G(j\omega)|$ defined as

$$|G(j\omega)| = \sqrt{\frac{1}{1 + \omega^2\left[\left(\frac{T_A}{k_P i K_A}\right)^2 - 2\left(\frac{T_A T_\sigma}{k_P i K_A}\right) + \omega^4\left(\frac{T_A T_\sigma}{k_P i K_A}\right)^2\right]}} \tag{3.7}$$

was equal to one over the whole spectrum i.e.: $|G(j\omega)| \equiv 1\ \forall\omega$. This, however, is obviously not achievable. Thus, the demand must be that the frequency response is equal to one for as many frequency values as possible starting from zero. The same as above can be expressed as

$$\frac{d^n}{d\omega^n}|G(j\omega)| = 0,\ n \in \mathbb{N} \tag{3.8}$$

while maximizing the order of derivatives, i.e. $n \to \infty$ for which equation 3.8 holds true. Thus, we receive the optimal setup for $k_P i$ with respect to the statement above as

$$k_P i = \frac{T_A}{2T_\sigma K_A} = \frac{T_A}{2\left(T_t + T_G i\right)K_A} \tag{3.9}$$

Substituting 3.9 into 3.6 results in

$$G(s) = \frac{1}{1 + s \cdot 2T_\sigma + s^2 \cdot 2T_\sigma^2} \tag{3.10}$$

If one compares 3.10 to the standard form of a PT2 element given as

$$G_{PT2}(s) = \frac{k}{1 + s \cdot 2dT + s^2 \cdot T^2} \tag{3.11}$$

the overall time constant T and damping factor d can be computed. We receive $T = \sqrt{2}T_\sigma$ $d = \frac{1}{\sqrt{2}}$. As stated in [Pfaff 94], the acquired damping is ideal in a sense of maintaining a phase response of one as long as possible. Notice that the complexity of the system has been reduced to only one free parameter that represents the summed time constant for the actuator and the controller delay.

3.2.2 Speed Control Loop

Since one of the foremost important tasks is to keep the resulting transfer function as simple and manageable as possible, the inner control loop can be approximated using a PT1 instead of a PT2 element without noticeable loss of accuracy. Hence, the time constant of the PT1 element is set to $T_{Si} = 2 \cdot T_\sigma$ as stated in [Pfaff 92, p.53]. Furthermore, the modeling of the outer loops includes the PI speed controller as well as the optional delay element representing the change rate of the reference input.

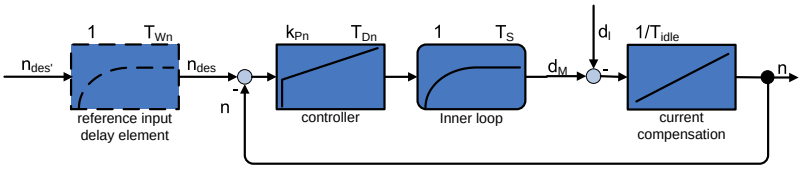


Figure 3.3: Structure of the outer control loop

Again, some parameters have to be determined in order to deduce the transfer function of the loop. First of all, the PT1 element time constant (i.e. inner loop delay) can be calculated as the sum of the delays of the smoothing element T_{Gn} and the controller/plant delay T_{Si}. Setting $T_{Gn} = 10^{-3}\ s$ (DSP control cycle time) and $T_{Si} = 4 \cdot 10^{-3}\ s$ based on the performance available in the discrete time current controller implemented in section 3.3 in order to receive comparable results leads us to

$$T_S = T_{Si} + T_{Gn} = 5 \cdot 10^{-3} s \tag{3.12}$$

Hence, we receive the open loop transfer function of the inner circle as

$$G_{0n}(s) = k_{Pn} \frac{1 + s \cdot T_{Dn}}{s \cdot T_{Dn}} \frac{1}{1 + s \cdot T_S} \frac{1}{s \cdot T_{idle}} \tag{3.13}$$

Since the plant is controlled by cascaded controllers, the prerequisite for such a structure, considering that the inner controller has to be significantly faster than the outer one, has to be fulfilled. Hence, $T_{Dn} > T_S$ must hold true.

In order to find the best setup for the proportional gain k_{Pn}, one has to analyze the Bode diagram of the open loop as suggested by [Pfaff 92, p.68]. Doing so, one can conclude that an optimum can be found in case both break points $\omega_1 = \frac{1}{T_{Dn}}$, $\omega_2 = \frac{1}{T_S}$ are symmetric to the gain crossover frequency ω_D in respect to the value characteristic. Thus, a maximum phase margin can be achieved which results in the most stable system behavior. Therefore, we receive

$$\log \omega_D = \log \omega_{Dn} + \frac{\log \omega_S - \log \omega_{Dn}}{2} \Leftrightarrow \omega_D = \sqrt{\omega_{Dn} \omega_S} \tag{3.14}$$

Since around ω_D only the proportional gain and the constant of the I-controller contribution are of mentionable effect, the transfer function can be significantly simplified for the purpose of finding a setup for the proportional gain

$$|G_{0n}(s)| = \frac{k_{Pn}}{T_{idle}\omega_D} = 1 \tag{3.15}$$

Substituting 3.14 into 3.15 enables us to find an expression for the proportional gain

$$k_{Pn} = \frac{T_{idle}}{\sqrt{T_{Dn}T_S}} \tag{3.16}$$

Since different specifications concerning the command response and disturbance rejection are to be met, it becomes useful to take a direct look at both. This is because it is hard, if not impossible, to solely rely on the Bode diagram for the entire design process. A good setup that meets both requirements is found when the so called 'symmetric optimum' is used. It demands the ratio between T_{Dn} and T_S to be $T_{Dn} = 4T_S$. As one can see, this only specifies the earlier mentioned cascaded controller requirement a little more precisely. Making use of this ratio in equation 3.16 leads us to

$$k_{Pn} = \frac{T_{idle}}{2T_s} \tag{3.17}$$

Thus, we receive the closed loop transfer function as:

$$G_n(s) = \frac{1 + s \cdot T_{Dn}}{1 + s \cdot T_{Dn} + s^2 \cdot \frac{T_{Dn}T_{idle}}{k_{Pn}} + s^3 \cdot \frac{T_{Dn}T_{idle}T_S}{k_{Pn}}} \tag{3.18}$$

Compensating the system's D-component (numerator zero) using the already familiar reference input delay element and substituting equation 3.17 as well as the symmetric optimum estimation into equation 3.18 leads us to the simplified system's closed loop transfer function:

$$G_n(s) = \frac{1}{1 + s \cdot 4T_S + s^2 \cdot 8T_S^2 + s^3 8T_S^3} \tag{3.19}$$

As seen before, the system's complexity could be reduced to only one free parameter.

3.2.3 Closed Loop Performance and Stability Analysis

In order to take a concluding closer look at the control system, an in-depth stability analysis becomes inevitable. It is even more important when we realize that the controller actually used to control the real hardware later on will be time discrete. This means that the familiar means of investigating the stability of LTI systems cannot be applied. Hence, a fundamental stability discussion will be performed for the control system deduced in the preceding paragraphs in order to transfer the results to a time-discrete system. The transfer of the qualitative stability analysis results is feasible in this case since, as shown later, both the time-continuous and the time-discrete system show an approximately equal system behavior. This point will be investigated closer in paragraph 3.4.

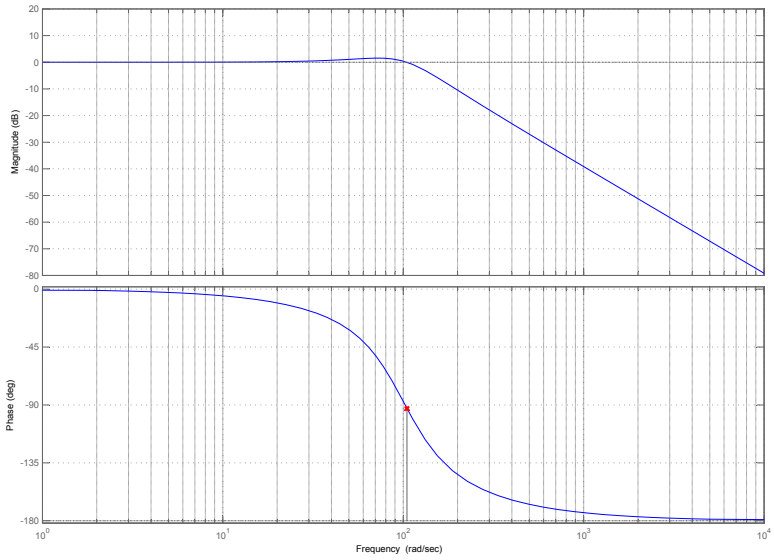


Figure 3.4: Bode plot of closed control loop

Concerning the study of the overall stability, the Bode plot seems to be best suited. As one can conclude, the presented system is a minimumphase system. Accordingly only one zero at $z_1 \approx -90.91$ and three poles, all located in the negative open half-plane ($p_1 \approx -100$, $p_{2,3} \approx -50 \pm i \cdot 86.60$) can be found. Thus, the system can be assumed to be stable by definition. Another fact underlining this point is the phase margin. As one can see in figure 3.4, it is approximately $+87.30\ deg$ (at approx. $105\ rad/sec$). Therefore, the system is stable according to the phase margin criterion ([Lunze 07, p.414]).

In order to be able to characterize the behavior in the time domain (i.e. command response and disturbance rejection), the frequency domain is not of use. For this purpose, the system was probed using two different input signals. The step response illustrates the system's command response. Therefore, the system output recorded in response to a step-function like change in the reference input is shown in figure 3.5. As one can see, the system is able to steadily reach the specified speed within approximately 60 ms delay. Furthermore, a short 18% overshoot can be observed that is most likely caused by the not entirely compensated D-component of the system's transfer function. Again the stability is underlined by the fact that the beginning oscillation is rapidly decaying.

The system-inherent disturbance rejection can be observed best when it is stimulated with the Dirac delta function. The observed system behavior is shown in figure 3.6. As one can see the initial reaction with a massive overshoot is dying out quickly and the previous system output is reached and maintained after roughly 95 ms with an error level of $\pm 10\%$.

Now that both the deduction and the analysis of the continuous system is performed, the attention has to be directed to the design of the time discrete controller. Although the immanent diversity of those two domains prohibits the direct transfer of the results found in the time-continuous world, the basic design ideas and results can be utilized anyway.

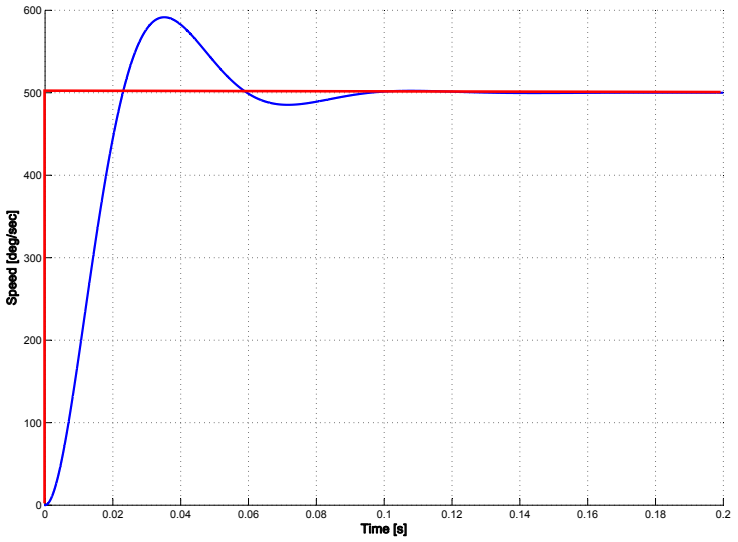


Figure 3.5: Response (blue line) to step-like change in reference input (red line) from 0 to 500 deg/s

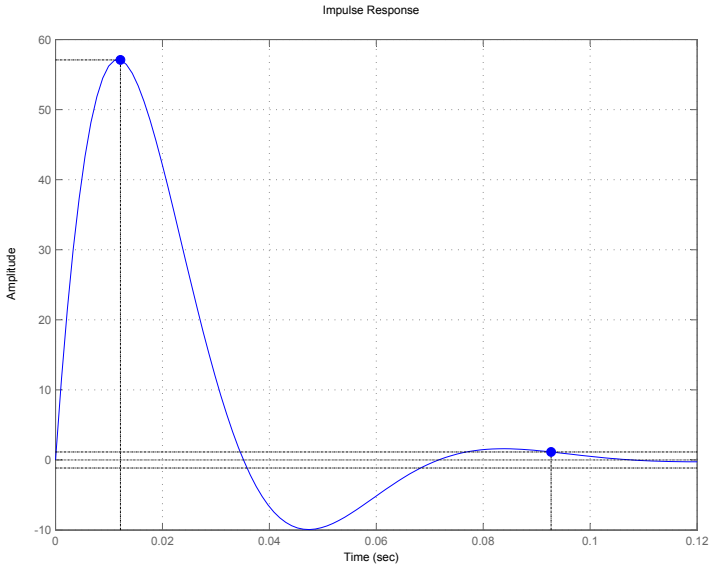


Figure 3.6: Impulse response of the closed loop illustrating the disturbance rejection

3.3 Time Discrete Modeling

Now that the time-continuous control system is designed and tested, it is time to move the focus to the actual controller that will be used on the hardware. For this purpose, the domain is redefined while trying to pursue the same approach in the time discrete world as done before for the continuous case. Using the results already acquired is the most convenient and reasonable since this allows for transferring of the concepts that have been already proven useful in the above case. Hence, in the following paragraphs both the adjustments in the way the control algorithm is implemented as well as the environment required for the experiments with the controller are discussed. The latter is needed to be able to assess performance and stability aspects on the computer without putting the real hardware in jeopardy.

3.3.1 Controller Implementation

As a starting point for the threefold cascaded (position, speed, torque) controller, the configuration suggested in [Hillenbrand 06] was expanded in order to be able to cope with the new challenges in form of the jumping task. One has to point out that the overall controller structure only had to be modified slightly. In order to be able to adapt to the new demands in terms of response time, precision, and features, the following setup has shown promising results during the simulation process.

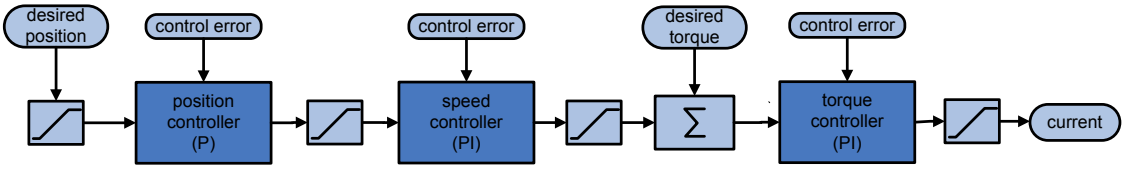


Figure 3.7: Schema illustrating the working principle of the time discrete control algorithm implemented in the DSP

As one can see in figure 3.7, the upper control branch (position and speed controller) was almost exactly transformed from the C++ code into MatLab, although a few modifications concerning the limiting devices had to be made. This is caused by the change in structure. The previously innermost controller (speed) is now cascaded by the current controller below. The most inner loop has become necessary in order to be able to deal with the commands of the behavior based control architecture in an appropriate way. This is because not only positions but also torques have to be produced in the respective joints now. Since the torque is directly proportional to the current, this can be realized using a current controller.

The feature was not included in the original controller because it was intended for controlling indoor robot drive motors. For that purpose, a torque control was never really needed. Now that the task is to perform a jump motion on a mechanical leg, however, this is unavoidable to provide an active compliance-like behavior.

The overall idea of the control schema is that the preceding controller generates a portion of the input for the next controller and is thus able to influence the acquired joint parameter setup. Each controller (except for the first in line) receives three inputs that influence the respective output. Those inputs are the already mentioned outputs of the preceding controller as well as the actual parameter value (e.g actual

current) and the respective desired value set by either the user or the control architecture. In this configuration the speed controller is given a special role to play. It is the only controller that is not directly influenced from outside of the control algorithm since neither a desired value is set nor is a weight for the later on fusion given. It is just designed to serve the position controller as a 'translator' for the needed change in speed to cause a change in position. So one may speak of a 'slave' controller if you like.

The already mentioned limiters are in this case means to fulfill the requirements for a cascaded controller. This way, one is able to ensure that the rate of change in the desired parameter value is limited by the dynamic performance of the next inner controller. In other words, the first controller is only allowed to change its output at a rate that allows the next inner controller to first acquire a control error below a specified threshold for the previously demanded value before a new demanded value can be set. Here, this is done by the relative limiters implemented in between each of the inner controllers. This way a maximum change rate for a value v is specified (i.e. Δv per control cycle). The absolute limiters are intended to prevent the integrator windup from happening (in between controllers) as well as to protect the hardware from maloperation by providing desired values outside of the acquirable limits or limiting the resulting current to a level that is not harmful to the armature windings. This prevents the thermal destruction of the motor.

Since the controller alone is just a means of executing commands that are given by a higher level 'intelligence' (i.e. the robot's control architecture), a command interface needs to be established. It can be denoted as a command vector $\vec{d}$ comprised of four components:

$$\vec{d} = (desired\_position, desired\_torque, weight\_position, weight\_torque) \tag{3.20}$$

The command interface described above enables the control architecture not only to transmit the desired values for both position and torque but also a respective weight indicating the importance of the respective reference input. Thus, the controller must be equipped with means to interpret the command vector in a way to generate a matching setup of joint parameters. This can be achieved by performing a fusion of the torques demanded by the speed controller and the torque controller. In mathematical terms, the resulting current (i.e. torque) is calculated as a weighted sum of the summands

$$current = \frac{weight\_position^2 * torque_{speed} + weight\_torque^2 * torque_{torque}}{weight\_position + weight\_torque} \tag{3.21}$$

Therefore, the time discrete controller implementation is now complete and the simulations used to find good values for the individual controller gains and to judge on the controller performance can now be performed.

3.3.2 Simulation Environment

In order to be able to verify the expected performance of the controller, a simulation environment had to be created that allows for measuring all needed system (i.e. controller and motor) parameters. Other aspects in the process of choosing the best suited simulation environment are easy usability and high simulation performance in respect to both speed and precision. Thus, MatLab Simulink was selected because it offers a vast range of functionality and is well established for scientific applications.

The first step in the design process is to deduce a proper way to represent the plant (i.e. motor) of the control loop. Therefore, a standard model for DC motors that can be found throughout control theory literature (e.g. [Foellinger 08]) was utilized and complemented with the motor specific parameters, taken from the manufacturer's data sheet. The result is presented in 3.8. This way a sufficient precision can be achieved for the initial test.

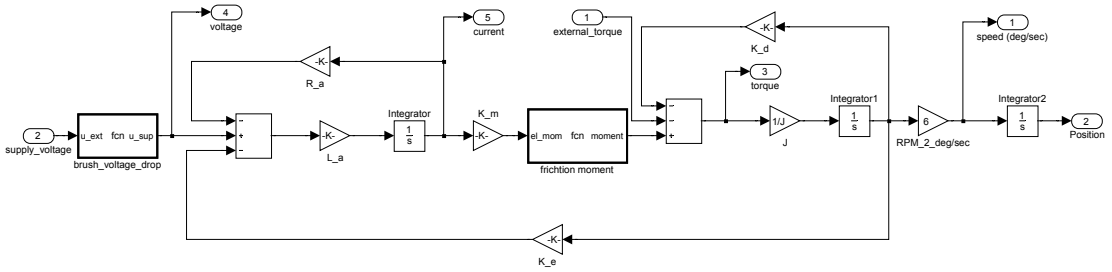


Figure 3.8: MatLab Simulink model designed to emulate the motor used in both the hip and the knee joint

As one can see, the LTI model was extended using two non-linear blocks (brush voltage drop, friction momentum) in order to receive a more realistic impression of the results that can be expected on the real hardware. Once the modeling of the key element was completed, additional components had to be incorporated into the simulation environment. A high level overview of all used modules is illustrated in figure 3.9. Notice that each block represents a complete sub-system that is refined in several stages to provide the required functionality.

Besides the already discussed motor, the main elements used in the simulation environment are the DSP block, a block that discretizes the encoder values and data acquisition blocks. The DSP block consists of a trigger unit to emulate the control cycle delay of 1 kHz $= 10^{-3}s$ and the controller implemented as MatLab m-file function block. The discretization is needed to emulate the real hardware behavior since the exact internal states (like e.g. the speed) cannot be measured in a floating point number but rather can be approximated with the encoder ticks measured (i.e. an integer number) in each control cycle. The data acquisition units are needed mostly for the setup process for debugging purposes. Only values that will also be accessible in the real hardware are used as input variables for the controller block in order to maintain a realistic concept.

3.3.3 Influence of PWM

In order to produce meaningful results in respect to the real hardware during the simulation process, it is important to consider the sources of deviation of the real out-

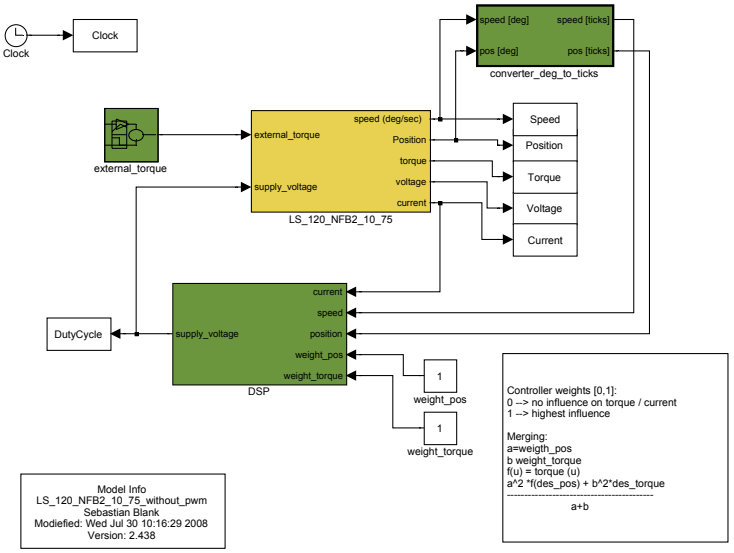


Figure 3.9: Simulink simulation environment used to simulate the performance of the controller in order to control the motor

put in respect to the simulated. This task turns out to lead to a decision concerning the acceptable deviation level with a trade-off between precision and performance.

A major factor concerning the simulation performance is the specified step size. It stands for the delay between two samples where all system parameters are recomputed or in other words the granularity or resolution of the result. The dominating factor in determining the initial step size is the parameter with the highest change rate. In this case it is the PWM output of the controller. It operates at 39 kHz which leads to a cycle time of approx. $2.6 \cdot 10^{-5}$. In order to ensure a minimum resolution of 10 samples per period, the simulation step size has to be set to 2.5 μs. Since experiments have shown that the duration of the simulation computation grows exponentially with the sample time, investigations on the impact of the overall results of the PWM became inevitable.

Figure 3.10 shows the results that were found during this process.

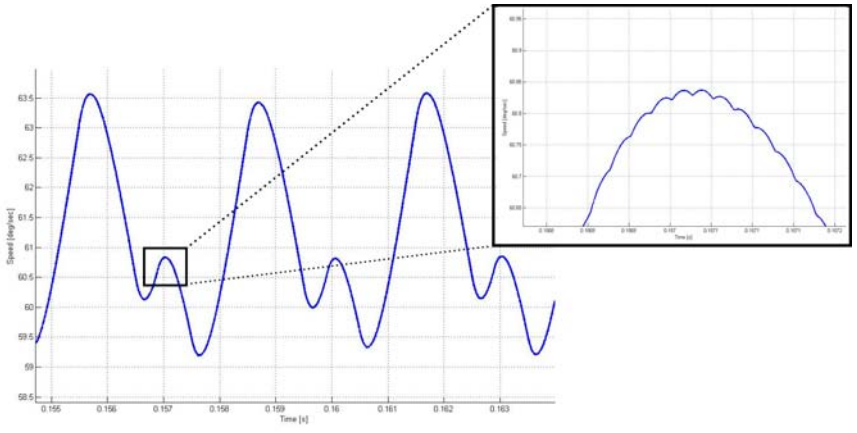


Figure 3.10: Influence of DSP cycle time relative to PWM impact

As can be seen, the impact of the change in output caused by the PWM (zoomed plot on the right) is approximately 0.01 *deg* while the impact of the DSP cycle time (oscillation in the left plot) is a little over 4 *deg*. Considering the error ratio of 400:1 and the significant loss in simulation performance[3] it became obvious that the PWM can be neglected in this context. Thus the system complexity and simulation time could be dramatically reduced.

3.4 Simulation Results

Now that the idea behind the controller and the respective design process has been exhaustively discussed, it is time to summarize the results that can be achieved using the presented approach. It is, however, indicated to first review the transferability of the stability analysis (performed for the LTI-controller earlier in this chapter) to the found time discrete controller to ensure overall system stability before taking a look at the specific results. Since the equality of the two systems cannot be systematically proven here, the most obvious way to underline the homogeneous behavior is to take a closer look at both the time domain reaction (like e.g. the step-response) and the frequency domain characteristics in form of e.g. the Fourier Transform of the system's reaction to a change in reference input.

In figure 3.11 the command response (speed step size: 500 deg/sec) of both systems is depicted.

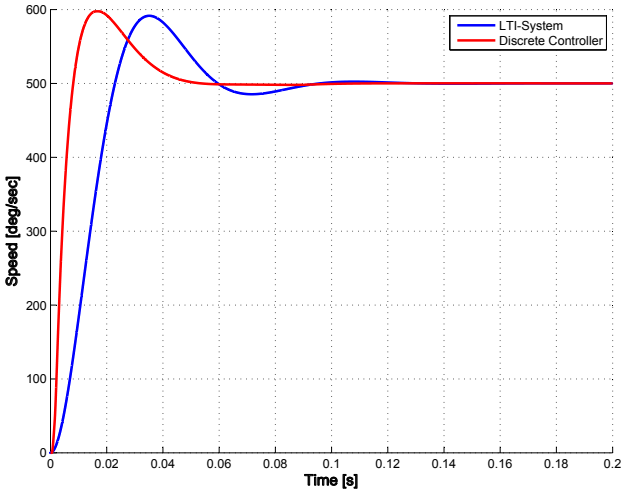
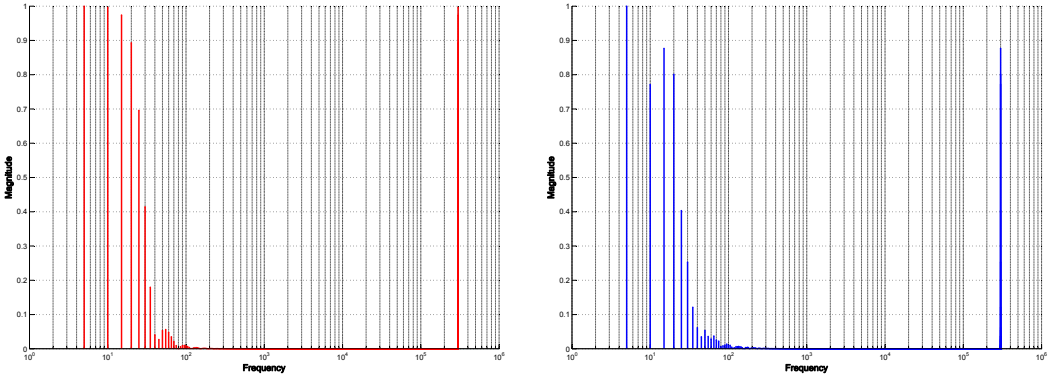


Figure 3.11: Command response of the time-continuous closed control (red) and the performance of the time-discrete controller (blue)

It is obvious that the time continuous LTI system produces an output that is very similar to the one that can be observed in the time discrete system. The two characteristics that distinguish both systems are the slower overall response and fade characteristics of the oscillation that can be observed in the LTI system compared to the latter. This is most likely caused by the more aggressive parameter setup necessitated by the discrete nature of the system and the introduction of limiters

[3]Simulation of 1 second (step size 2.5μs) of system output takes about 45 minutes on a regular desktop computer.

that cannot be found in an LTI system because they resemble a non linearity. Thus, it can be stated that both systems show an overall equal time-domain behavior. This alone, however, is not sufficient to assume equal system characteristics in respect to stability. Thus, a closer inspection of the frequency domain properties was performed as illustrated in figure 3.12.



(a) Continuous system **(b)** Discrete system

Figure 3.12: Normalized Fourier transformed impulse responses.

Both systems were stimulated with a sudden change in the speed respond command of approximately 600 deg/sec. Their output was recorded for a simulation time of 0.3 seconds and afterwards Fourier transformed. After that, the magnitudes of the frequency spectrum were normalized. This allows for better visibility of the spectrum aside of the main peaks. Again, one will find that it is safe to say that the two spectra show overall equal characteristics. Those are the two dominant peaks at the beginning and end of the spectrum[4] indicating a strong constant component. The first peak is followed in both spectra by descending magnitudes until a frequency of about 50 Hz is reached. Furthermore, the presence of a repeating oscillation pattern can be found until a frequency of approx. 300 Hz is reached. This is adjoined by almost zero magnitudes until the second peak is reached at approx. $3 \cdot 10^5$ Hz.

In conclusion it can be stated that the transferability of the stability characteristics can be assumed. This is because the two systems show almost equal input/output behavior for various signals in both the time- and frequency domain.

With the stability question solved, one can now redirect the focus to the performance of the controllers in respect to the expected demands for the control task. Since the presented system demands multiple actuating variables[5] it makes sense to investigate the performance of each individual controller in the context of the entire control structure. Before this can be done, however, it is reasonable to first reflect on the range of the command variables. Preceding considerations about the required change in the variables for a jumping task led to the following:

- Position: Considering the human as biological role model for a jump, a fitting estimation can be given as approximately ± 90 deg per joint and second.

[4] approximately $10Hz$ and $3 \cdot 10^5 Hz$

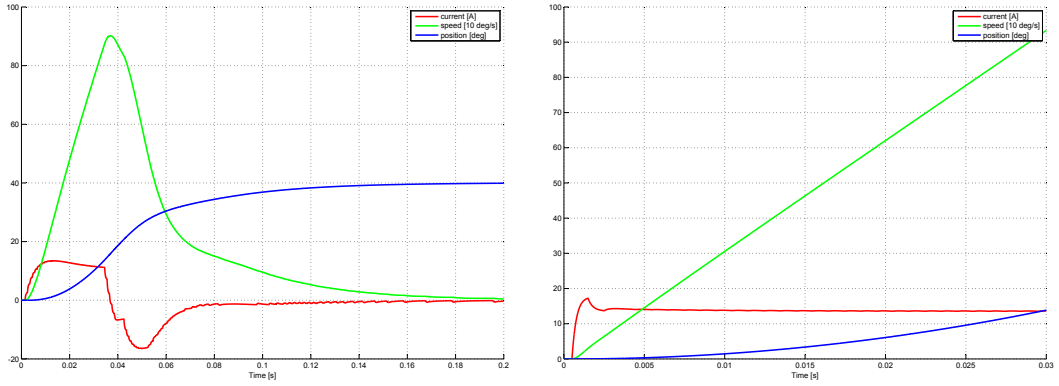
[5] i.e. position, speed and torque

This allows for a change of 45 *deg* and back into the point of origin which is considered to be sufficient for a jump including a certain amount of counter-movement to sustain a mean frequency of 1 Hz.

- Speed: In order to provide the mentioned change in position the speed controller has to be able to provide a satisfactory changing rate for the position i.e. speed. Experiments showed that a speed of ± 180 deg/s is adequate to guarantee the desired overall performance.

- Torque: As the innermost of the three cascaded controllers, the torque controller has to be the fastest in order not to slow down the entire controller. Thus the relative change in torque has to be sufficiently large. Since in DC motors torque is directly proportional to the motor current, it is convenient to express the torque in the form of a current. For the specific task, a change in value per second of ± 35 A has proven to be an optimal setup considering the reaction time of the plant and the limitations implied by the two controllers located above the torque controller in the cascaded structure.

Since in general those extreme values will only be achieved rather rarely, it is useful to consider a more typical value for the reflexion on the general controller performance. The result of this process is depicted in figures 3.13(a) through 3.14.



(a) weight_pos = 1, weight_torque = 0 **(b)** weight_pos = 0, weight_torque = 1

Figure 3.13: Performance of the simulated discrete time controller for response commands: position 40 *deg*, torque 15 A and fusion weights as specified below each figure.

Since the high level control architecture is equipped with an interface as described in the previous paragraph, the speed controller will not be addressed directly but is already included in the position controller performance. Thus, only tests for position- and torque commands are presented. In order to show the working principle of the weighted torque fusion that was already discussed, these parameters were altered according to the command values.

The command values for both the position and torque controller were kept constant during all three simulation runs while the weights were set as specified in the captions in order to allow the reader to study the impact of this values and to be able to judge on the controller performance at the same time. Figure 3.13(a) shows the

simulated controller output for a desired change in position. It takes the controller approximately 140 ms to induce the motor to reach the specified position. Although this may sound a little slow at first considering the task, it is more than sufficient. This is because this simulation is rather a worst case approximation of the real hardware in a meaning that the controller performance will significantly benefit from a decrease in the DSP control cycle time that can be performed if necessary. Under the given premises, however, the already acquired performance is more than sufficient for the task. The current and speed characteristics underline the docile overall behavior of the controller by the absence of noteworthy oscillations in either of the three recorded variables.

Now that the first case (a pure change in position) is discussed, let us redirect the focus to the torque controller. As one can see in figure 3.13(b), it is capable of adjusting to the desired current of 15 A within approximately 3 ms. This really fast reaction can be explained by the fact that this change in command variable only affects the innermost controller and can thus be performed rapidly. Again almost no oscillation can be observed.

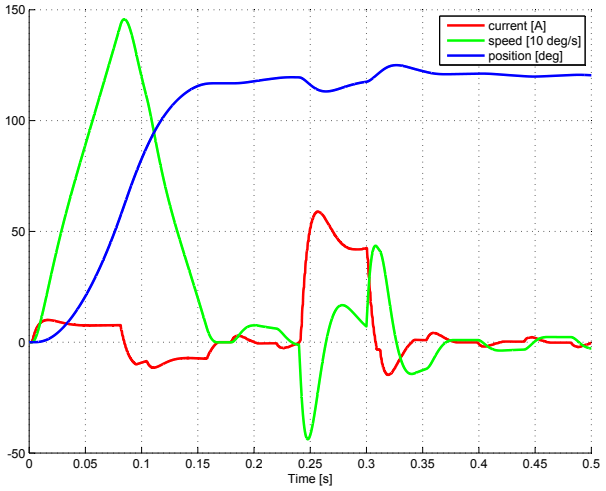


Figure 3.14: weight_pos = 0.7, weight_torque = 0

The last figure (3.14) illustrates the case of a 'soft' position controller, i.e. position weight 0.7 and torque weight 0. After the commanded position of 120 deg is reached, the plant is exposed to an external distortion at $t = 0.24$ sec. The desired behavior of a soft position controller is to be position accurate up to a certain degree and at the same time offer compliance when an external momentum is applied. Thus, the controller allows for a slight change in position before increasing counter momentum in an attempt to return to the desired position. After the distortion is withdrawn ($t = 0.3$ sec), the controller gradually decreases the momentum in order to return to its regular position.

4. Control Architecture and Dynamic Simulation

4.1 Modeling of the Leg

Before the work on the actual hardware can be performed, it becomes inevitable to first simulate the complex system in order to preserve the expensive setup from harm. A fatal event might be caused by something as simple as a faulty setup of the parameters or undetected errors in the code of either the controller or the control framework. Since the process is intended to reproduce the real world as closely as possible, a solid physics simulation framework is required. For this task the NEWTON[1] framework was selected. Another important factor that determines the degree of reproducibility in the real world is the modeling of the leg. The model of the prototype leg used for the simulation was taken from an earlier leg setup that was designed by Luksch et al. It is used for the initial investigations and later on altered based on the findings presented in this section. At first, the kinematic and dynamic properties of the leg will be discussed. Using a model that is as close to reality as possible is a very crucial point. Only this can guarantee that the results of the simulation process can be transferred to the real setup later on. Therefore, the already available leg prototype was entirely disassembled, weighed, and measurements were taken. This process resulted in the simulation model presented in the following paragraphs.

4.1.1 Kinematic Model

Previous to the discussion of the simulation process and the results in the successive sections, a kinematic model for the leg will be deduced. In order to do so, a fitting methodology has to be applied. For this cause the notation employing the Denavit Hartenberg (DH) parameter convention [Hartenberg 55] is used here. The graphical representation of the kinematic model used for modeling the leg can be found in figure 4.1.

[1] http://www.newtondynamics.com/

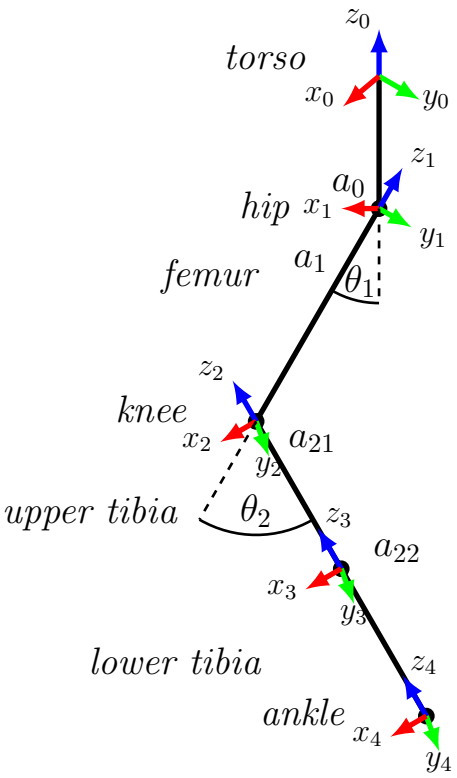


Figure 4.1: Kinematic model of the leg used for the simulation process

As illustrated in this schematic drawing, the leg itself is equipped with two rotational joints offering one DOF each. A load cell is installed within the tibia segment. This results in a separation of the upper and the lower part. Thus, a new coordinate system is introduced at this position for computational convenience. Furthermore, the leg is fixed to a slider structure above the hip. The purpose of this setup is to restrain the leg to solely move in positive and negative z-direction. Thus, the first coordinate system transformation from the WCS[2](x_0, y_0, z_0) to the hip coordinate system (x_1, y_1, z_1) can be obtained as follows: The distance between the hip joint pivot point and the fixture can be modeled using the proper matrix given in DH convention with the segment length a along the z_0 axis. Thus, we receive the transformation matrix A

$$ {}^{0}A_1 = T_{z_0}(a) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.1} $$

Notice the indices of the transformation matrix: The upper index in the front denotes the system of origin while the lower one indicates the frame the matrix is transforming the coordinates to. The first frame representing an actual leg joint is the hip coordinate system. Since the coordinate axis y_1 can be identified as the

[2]WCS: world coordinate system

joint's pivot axle and assuming the rotation angle to be θ, the transformation can accordingly be modeled as

$$R_{y_1}(\theta_i) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\theta & 0 & s\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

The notation in equation 4.2 was introduced for reading and writing convenience. Otherwise the complex matrices that will occur during the deduction of the kinematic model would become unpleasantly large. Thus, we can proceed to finish the modeling of this joint by composing the last matrix representing the translational offset caused by the segment length:

$$T_{z_1}(a_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

In order to receive the complete kinematic conversion from the WCS to the hip coordinate system one simply multiplies the matrices in the correct order:

$${}^{WCS}A_{hip} = R_{y_1}(\theta_i) \cdot T_{z_1}(a_1) = \begin{pmatrix} c\theta & 0 & s\theta & s\theta \cdot a_1 \\ 0 & 1 & 0 & 0 \\ -s\theta & 0 & c\theta & c\theta \cdot a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

The subsequent coordinate transformation matrices can be derived in an analog manner: The knee joint represents a very similar setup to the one above except for the fact that no displacement is present and therefore the respective matrix can be omitted. The remaining joint has no physical representation and can thus be modeled as a pure translation in the negative z-direction. Hence we receive the following matrices:

$${}^{hip}A_{knee} = \begin{pmatrix} c\theta_1 & 0 & s\theta_1 & s\theta_1 \cdot a_1 \\ 0 & 1 & 0 & 0 \\ -s\theta_1 & 0 & c\theta_1 & c\theta_1 \cdot a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

and respectively

$${}^{knee}A_{ankle} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.6}$$

The forward kinematics can now be derived by simply multiplying the transformation matrices in the respective order:

$$^{WCS}A_{ankle} = \prod_{i=1}^{3} \left(^{i-1}A_i \right) \tag{4.7}$$

Thus substituting 4.4, 4.5 and 4.6 into 4.7 leads to

$$\begin{pmatrix} c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2 & 0 & c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2 & (c\theta_1 \cdot s\theta_2 + s\theta_1 \cdot c\theta_2) \cdot a_3 + s\theta_1 \cdot a_2 + s\theta_1 \cdot a_1 \\ 0 & 1 & 0 & 0 \\ -s\theta_1 \cdot c\theta_2 - c\theta_1 \cdot s\theta_2 & 0 & c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2 & (c\theta_1 \cdot c\theta_2 - s\theta_1 \cdot s\theta_2) \cdot a_3 + c\theta_1 \cdot a_2 + c\theta_1 \cdot a_1 + a_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.8}$$

A solution for the forward kinematic problem can now be found by substituting the DH parameters that are denoted in the table below into equation 4.8

Segment	Length (a_i)	Link twist (α_i)	Angle θ_i	Displacement (d_i)
torso	50 mm	0	0	0
femur	467 mm	0	θ_1	0
upper tibia	267.5 mm	0	θ_2	0
lower tibia	187.5 mm	0	0	0

The final resulting transformation matrix expresses that the overall movement is constrained to the x,z-plane, i.e. $y = const = 0$. This is physically enforced by the fixture that allows only movement in the z-direction. Since both rotational joints have a common pivot axle (i.e. y-axis) the leg offers only two translation degrees of freedom.

4.1.2 Dynamic Model

The task of deducing dynamics for the leg is motivated by the need of an assessment of the required momenta in order to fulfill the jumping task. The general description of the relation between momentum, inertia, gravity, and angle is given by the so called *equations of motion*[3]:

$$Q = M(q) \cdot \ddot{q} + n(\dot{q}, q) + g(q) + R \cdot \dot{q} \tag{4.9}$$

symbol		dimension	
Q	:	$n \times 1$	general momenta
M(q)	:	$n \times n$	inertia matrix
$n(\dot{q}, q)$	:	$n \times 1$	centrifugal- and Coriolis force vector
g(q)	:	$n \times 1$	gravitational force
R	:	$n \times n$	friction matrix
q	:	$n \times 1$	angular orientation

with i being the index that represents the segment resp. joint number. Deducing a proper model with the mechanical setup in mind is crucial in order to produce meaningful results during the simulation process. For this purpose, the setup presented in

[3]In physics the equations of motion describe the behavior of a mechanical system in respect to e.g. motion as a result of an externally applied force. An example for these equations is e.g. Newton's second law.

figure 4.2 was derived. The fact that the presented system is a redundantly actuated parallel kinematic chain makes the task of inquiring the dynamic properties not easy at all. The standard methodology like e.g. Lagrange's equations cannot be applied here due to the specific design of the system. In fact, the problem of inverting the dynamics for a limb chain is generally unsolved yet as stated in [Kalveram 91]. Thus, the general approach is to introduce simplifications that make the problem solvable. Instead of changing the character of the system, the dynamics were investigated using the free physics engine NEWTON. It is connected to the behavior based control architecture that will be presented in section 4.2.

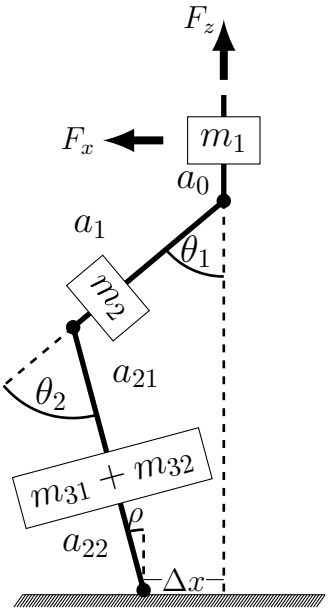


Figure 4.2: Graphic representation of the leg model suited for dynamic simulation

hip joint: (incl. mounting)	m_1	=	4.5 kg	a_0	=	50 mm
femur:	m_2	=	5.4 kg	a_1	=	467 mm
upper tibia:	m_3	=	0.8 kg	a_2	=	267.5 mm
lower tibia:	m_4	=	0.8 kg	a_3	=	187.5 mm
total:	mass	=	15.5 kg	length	=	972 mm

The task of the overall process is to analyze the dynamic model. In this particular case only the assessment of the approximately required torques is of interest. Therefore, the computational load can be significantly lowered by assuming a single mass point rather than a structure with distributed mass. In order to retrieve reasonable precision, the location of these points relative to the segment length is determined with respect to the mass distribution in the real hardware. Thus, the hypothesis in case of the tibia segment is that the mass of both the lower and upper segment is concentrated in the middle of the summed lengths. Thus, $a_2 = a_{21} + a_{22}$ and $m_3 = m_{a31} + m_{32}$ holds true.

The leg is intended to perform a jump of only $h = 10cm$ peak height. The minimum lift-off velocity that has to be obtained for this can be determined as shown in equation 4.10.

$$E_{kin} = E_{pot} \Leftrightarrow v_{lift-off} = \sqrt{2 \cdot g \cdot h} \tag{4.10}$$

In order to jump, the energy required for a change in height of the leg's mass (i.e. potential energy) has to be entirely taken from the kinetic energy at the moment of lift-off. Thus, the velocity at the peak point can be determined by equalizing the kinetic and potential energy. Rearranging the found equation allows for calculating the required velocity. In the table below, the required lift-off velocity is given for various jump heights

acquired peak height $[m]$	0.01	0.05	0.10	0.20	0.50
required velocity $\left[\frac{m}{s}\right]$	0.4429	0.9903	1.4005	1.9806	3.1316

For interpreting purposes of the results produced during the simulation process, it is useful to first determine the maximum acquirable torque using the motor and gear transmission included in the hardware. According to the data sheet, the zero motion torque is given as $M_{st} \approx 13.72\ Nm$. Using a transmission ratio of $32 : 1$ while neglecting frictional loss and so on, the maximum obtainable torque is given as $M_{max} = 439\ Nm$. Since a very high current ($> 100\ A$) is connected to this momentum, it is, however, not advisable to use this momentum for more than a few fractions of a second. Thus, the sustainable torque without having to put up with possible damage to the hardware can be assumed as approx. $M_{sus} = 150\ Nm$.

Based on the above findings, one can make several assumptions that will simplify the process of finding reasonable command values deduced from the mechanical structure. At first, it seems reasonable to try to keep the foot point and the hip joint more or less aligned in order to minimize lateral forces as well as slip on the surface. Slip can occur if the foot point is too far away from the projection of the hip's pivot axis to the ground plane. It can be observed if the surface friction is too low to ensure sufficient contact between the ground and the foot in order to transmit the entire leg momentum. The most critical situation is the lift-off phase. This is because high momenta are applied for only a short period of time in order to reach a sufficient lift-off velocity. A second very useful and evident rule of thumb is that, due to the restraints and lever, the major share of the momentum will be present at the knee joint. Thus, the lion's share of both the impact and lift-off stress is burdened onto the knee.

4.2 The Behavior Based Control System

4.2.1 IB2C - A Software Framework for Behavior-Based Robots

As mentioned before, the task of this thesis is to introduce a biologically inspired control architecture capable of controlling a robot. In order to do so, one first has

to choose a general paradigm that seems to be the most promising in the problem domain. Therefore, let us take a look on how mother nature solves the control task. As presented in [Dillmann 04], electroencephalography (EEG) and positron emission tomography (PET) scans in animals have shown that certain actions always create activity in the same region of the brain and spinal cord. While dealing with the information transmitted to them, the active regions in the central nervous system (CNS) stimulate or inhibit activity in others. This process forms the final system reaction to the input. Thus, the use of a behavior based architecture (BB) is indicated since it adopts this exact principle that has shown enormous potential in animals and humans.

Behavior based architectures in general and the integrated behavior-based control architecture (IB2C) in specific offers a multitude of advantages over others: The main benefit is that the ability to function properly is not dependent on the correctness of one central world model. This is a consequence of the general principle of such architectures.

The main aspect is to reduce the overwhelming amount of global information to a problem domain specific representation that incorporates the locally relevant aspects only. During the processing task, the overall complexity in the system can arise from the collaboration of the various behavior modules. Distributed functional blocks should be favored over a complex monolithic block since the implementation becomes easier this way.

If all design guidelines are followed, this approach guarantees a robust final system. The challenge that has to be tackled in the design process is the coordination of those more or less simple behaviors by means of inhibition and mutual stimulation. Unfortunately, this kind of approaching a problem also causes the main drawback: It is hard to track down the source of potentially undesired system behavior. The IB2C architecture as described in [Proetzsch 08] stands out due to the fact that it imposes very few modeling constraints as opposed to many others like e.g. the subsumption architecture by Brookes [Brooks 86]. Therefore it is applicable to almost arbitrary problem domains to be conquered. The basic building block in the world of IB2C is the behavior module as depicted in figure 4.3.

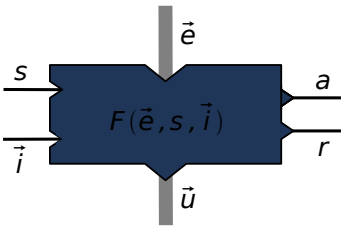


Figure 4.3: Template of a general behavior based module in the iB2C framework

It also defines a standardized communication interface in between those modules by means of *activity* and *target rating* as well as the possibility of *inhibition*. Therefore, the communication can be characterized as the flow of activity through the behavior network. In the perception of IB2C, an arbitrary behavior module B is defined by its target rating t, activity a, and transfer function F, i.e. $B = (r, a, F)$. Since the

task of a module is to provide a reaction to its current inputs and transfer function, the output u is acquired by applying the transfer function F on the input e. At the same time, the behavior's current *stimulation* s and *inhibition* i state have to be considered. In other terms, the output can be computed as $u = F(e, s, i)$. The complexity of the transfer function implemented in each block is in no way restricted. Thus, computations of arbitrary complexity are possible although it is advised to distribute complexity among several modules. This is indicated to retain the basic design idea behind BB architectures. An important paradigm expressed in the design of the communication interface is the separation of data and communication messages. However, in some situations it becomes necessary to merge certain data messages while considering the state messages (i.e. activity) of the involved modules. For this purpose, a *fusion module* is provided by the architecture that fills the described gap.

Before going into further detail with the fusion module, let us first redirect the attention to the conventions introduced for the state messages and discuss the meaning of the various terms involved in this process. This is of special importance because those terms and the values associated with them represent the basic means of communication: The stimulation $s \in [0, 1]$ of a module depicts the intended relevance of the behavior. It is provided by a module's predecessor(s) and determined in a way of maximum fusion. This means that the highest incoming stimulation determines the stimulation of the successor module. A general design rule allows any given module to relay as much as its own activity (in form of stimulation) to the modules, it is connected with. Before the activation can be investigated, it is reasonable to first introduce the inhibition in an example for a given behavior module B. This is because it is involved in determining the module's current activation. The inhibition $i \in [0, 1]$ of a behavior reduces its relevance. An inhibition message can be sent by up to k other modules where the resulting inhibition of B equals $i = \max_{j=0,\ldots,k-1} i_j$. It is designed to have the inverse effect of stimulation, i.e. it constrains the activation. The module's activation can be computed as the product of the cumulated incoming stimulation and the inverse of the cumulated incoming inhibition: $\iota = s \cdot (1 - i)$. The penultimate term involved in the process of exchanging status information is a module's *activity* $a \in [0, 1]$ which denotes the current influence of the behavior on the system state. A setting of $a = 1$ refers to full activity and thus the highest impact while $a = 0$ indicates complete passivity when it comes to impact on the resulting system behavior. An important design criterion that has to be met is that the degree of activation limits the module's activity. If expressed in mathematical terms, $a \leq \iota$ always holds true.

In order to allow for a true exchange of messages (i.e. bidirectional communication), a means of providing feedback of the module's current state is still missing. Hence, the *target rating* $t \in [0, 1]$ is introduced to express the content of an individual behavior with the current system state. In this context $t = 1$ indicates full discomfort while $t = 0$ equals full satisfaction.

Now that the overview of the basic vocabulary of the IB2C architecture is given, the introduction of further design principles is still imminent. The first principle signifies that the activity remains constant if both target rating and activation remain steady. This means that if no change in the input of the module occurs, the activity will not

be altered either. This is straight forward and well known in every day life like e.g. physics. The second principle is of equal importance. It states that the activation of an arbitrary module may not be affected by its target rating. The idea behind this design guideline is that the content of the respective module with the system state may not be dependent on the impact the module is having on it. In other words this can be rephrased to: The content with the state may only be dependent on the state itself.

Now that the communication principles are dealt with, one may refocus on the fusion behaviors. The usage of the latter is indicated in case of competing behaviors (in respect to an output, e.g. commands to the actuators) are present. The basic idea behind fusion in the context of BB modules is that a module with a high activity should have a larger influence on the output than another one with lower activity. The ability to realize a variety of imaginable applications implies more than just one way to fusion outputs. Therefore, two basic methods are implemented so far: maximum fusion and weighted fusion. In the former case only the most active behavior gains influence on the fusion's output, while in the latter case the output can be determined as $u = \frac{\sum_{j=0}^{p-1} a_j \cdot u_j}{\sum_{k=0}^{p-1} a_k}$.

4.2.2 General Structure

In order to underline the biological roots of the system, it makes sense to adopt even more aspects of the control architecture that can be found in both animals and humans. Therefore, the system ought not only to be based on a BB paradigm but also features a hierarchy that allows for the use of abstraction. In this context abstraction covers both the information about the environment and the system state (sensor data) as well as the type of command value. Such an approach allows for a gradual solution of the problem with an inclining level of abstraction the further one moves up in the hierarchy. Hence, each layer only has to deal with its specific scope of the problem and contributes its share to the overall system reaction. Again, the richness in the system's abilities arises from diversity. The fact that animals only use very sparse sensor information in the locomotion process underlines the importance of an approach that is well suited for the problem domain.

In order not to start from scratch when it comes to coordinating the multitude of BB modules, it seems reasonable to adapt an idea initially established for cyclic walking presented in [Luksch 08]. This can be done because the motion coordination functionalities for walking and jumping are fairly similar. The specific content, however, has to be adjusted to the new problem domain. Therefore, only the general structure remains unchanged. Before we go on with a more detailed presentation of the control structure, let us first focus on the environment used for the simulation process. The general structure for the simulation is illustrated in figure 4.4.

Figure 4.4: General structure of the simulation environment used

It is comprised of three pieces: the control layer, the hardware abstraction layer (HAL), and the physics engine. The HAL 'hides' the complexity of the hardware to the top layer by translating abstract commands into more complex ones that are understandable for the hardware. Since a bidirectional communication is required, the sensor information has to be processed as well before it can be handled up from the robot to the control architecture. The free physics engine NEWTON is used to emulate the behavior of the real hardware. Therefore, all relevant aspects of the actors, sensors, and all passive components are modeled into the engine in order to produce results as close to reality as possible.

The control layer itself represents the control architecture. Just like its biological counterpart, it is structured into three abstraction levels: brain, spinal cord, and muscle group. The brain represents the high level system coordination between the 'modes' the leg can be in. In case of this thesis this is either standing, transition into cyclic jump, or repetitive jumping. This is achieved by the implementation of so called skills. Each skill is responsible for taking care of a specific phase of the motion. In accordance with the high level, the sensor information representation is very abstract and sparse. The command interface to the lower levels is designed the same way. As stated above, the brain more or less only determines the general situation of the system and reacts to that information by activating the respective modules on the lower level. It has to be pointed out that it does not directly assign e.g. position commands. This is done in order to stay consistent with the design principles. Its main response is to act as a state machine containing the current system 'mode' and managing the transitions between these modes.

The layer located below the brain is the spinal cord. Its responsibility is to translate high level decisions into commands for the lower layer and to coordinate them by routing stimulation to the respective muscle group module. Again, the purpose of this level is mainly to coordinate, but on a different abstraction level. Its main feature is a module coordinating the reflexes on the lower level during the jumping task. It also accommodates modules generating the command patterns for the mode transition and stand phase. As mentioned, the abstraction level is lower as before. Thus, the actor interface allows for the commanding of desired angles for each actor. As a matter of fact, these commands do not reach the actual hardware directly but are fed into the lowest layer: the muscle group.

This group contains the actual reflexes responsible for the actuation itself. A reflex can in this context be understood as a tight actor sensor coupling. This mechanism gets rid of the need to ascent the whole way to the top level and back down again before a reaction can be performed. This ensures a feature of major importance: low latency between the occurrence of an event and the proper response to it by the control architecture. For coordination purposes, the reflexes may be inhibited by higher level's skills. This, however, is dependent on the current system state. Of course this concept is also enforced when dealing with sensor information. The reflexes have access to almost the full sensor information gathered, only slightly processed by the HAL. The hardware interface of this layer features the full spectrum of commands. Therefore, torques matching the desired hardware behavior are generated in addition to the angles. Before being fed into the hardware, the commands

are passed through a separate fusion module. Each one is associated with either actor in order to generate the respective output.

4.2.3 A Behavior Network Capable of Performing Repetitive Jumps

Now that the structure of the control block is derived, it is time to take a close look at the actual content of each layer (i.e. group). Figure 4.5 depicts the more detailed structure of the architecture's collaboration principles. Each level is represented by an individual bounding box. All blue components are realized within the IB2C framework while the elements kept in red are about to be implemented in the HAL and the physics engine. The fusion modules are indicated by the circles labeled with the sum symbol 'Σ'.

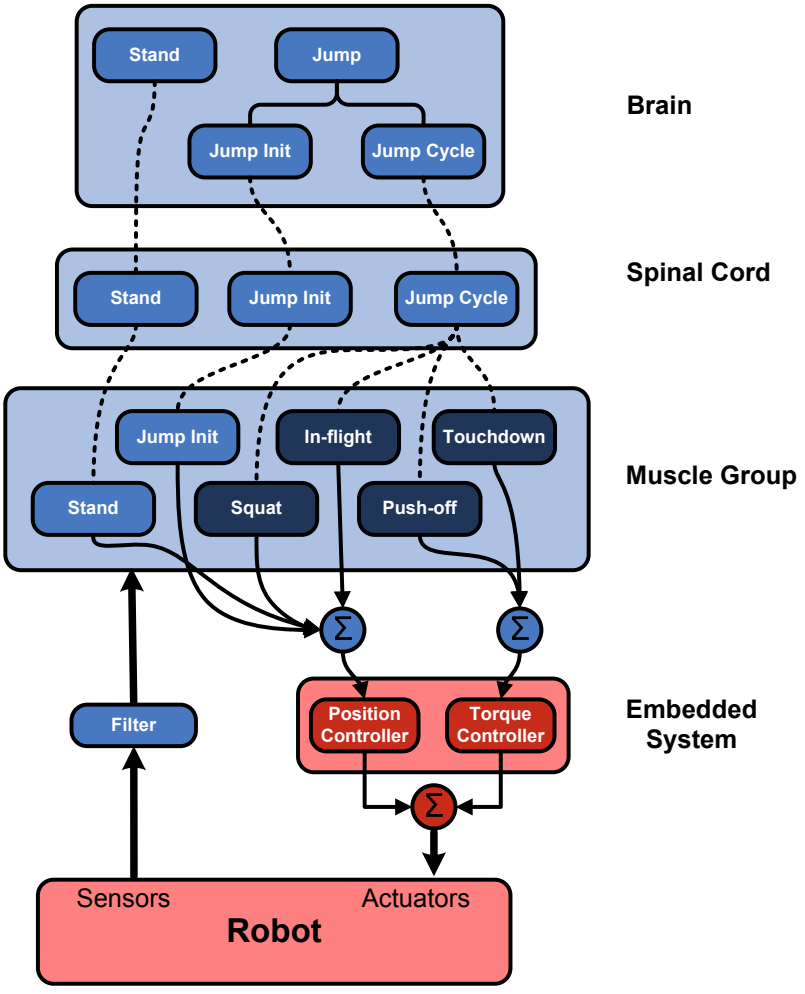


Figure 4.5: Schematic overview of the behavior network controlling the motion of the leg

For this purpose, several design principles were introduced prior to the implementation of each module. This is intended to ensure the desired overall result. The first one is based on the 'divide and conquer' principle enforced throughout the architecture. It states to keep each transfer function easy. Instead, functionality should be spread over several behavior modules. The reason for this is quite obvious when

the earlier specified demands towards this architecture are reconsidered. The second and equally important one is to favor torque control over position control wherever possible. This has two effects, that are desired in this project: On the one hand, it allows natural dynamics to unfold and contribute to the effort. On the other hand, this guarantees a certain amount of inherent robustness when it comes to external distortions. Now that the basic schematics are discussed, one may go on with investigating the reflexes themselves. Before we do so anyway, it is important to first survey the information processing in animals to understand the overall process. This is indicated to allow for designing reflexes in accordance to the found results.

As pointed out in [Pearson 95], the main sources of information for a locomotion task in animals are proprioceptive[4] organs. Those are e.g. stretch receptors in neuromuscular spindles located in muscles. They are used to determine forces as well as angular positions. In other words, the sensor information involved in the natural motion processes is pretty sparse. Inspired by that, four basic reflexes were designed in order to coordinate each phase of the jump motion. Those are the push-off-, in-flight-, touchdown- and squat reflex. Each single one is self-dependently controlling a phase of the motion. The correct coordination of the reflexes results in the ability to perform repetitive jumping motions. Their temporal order is expressed in figure 4.6.

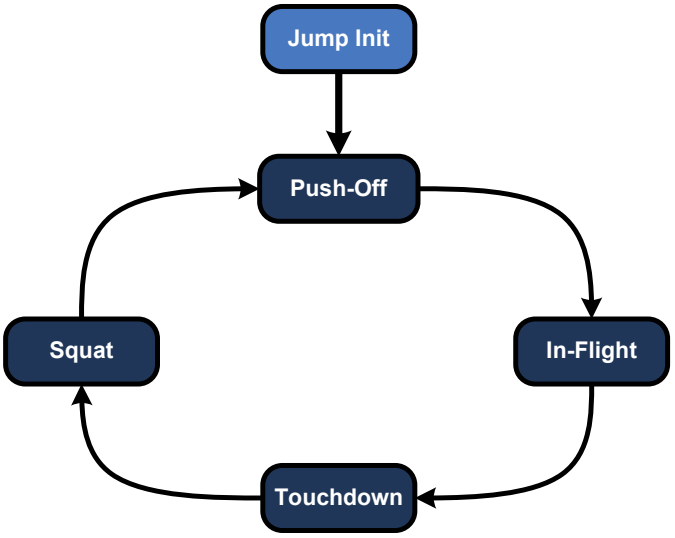


Figure 4.6: Reflex sequence during a jump neglecting the concurrency effects

4.2.3.1 Push-Off Reflex

The push-off reflex is intended to start in a squatted position. This is ensured by an initially active spinal cord level skill (see previous section) and the squat reflex, described in one of the following paragraphs. Once the reflex is stimulated, the leg is stretched out by applying torque to either actuated joints. Experiments pursued during the implementation phase have shown that the naive approach of applying the maximum momentum at both joints is not to be favored. In fact, it is contraindicated

[4] from *lat.* *proprius* one's own and *perception*: sense of relative position of neighboring body parts.

because this would induce undesired lateral forces due to the closed kinematic chain of the leg. Thus, the hip is deliberately relaxed almost entirely while the main share of the work is performed by the knee actuator (knee 100 %, hip 5 %).

Torque is applied until the leg is close to the full extension but still remains slightly bent. It is entirely withdrawn after this point in time to reduce the lateral movement in negative x-axis direction after the lift-off. Besides eliminating undesired movement, this also helps to reduce the energy consumption of a jump motion. This is because the energy 'wasted'[5] during the push off will have to be compensated for by even more power invested into the counter movement while the leg is airborne. Once the foot point loses contact to the ground, the activity of the reflex is withdrawn by the coordination function located on the spinal cord level. It is then handed on to the reflex controlling of the leg while it is airborne. This is the so called 'in-flight' reflex.

4.2.3.2 In-Flight Reflex

The intention behind the in-flight reflex is the necessity to ensure a proper landing posture. This is required in order to minimize the mechanical stress on the joints and segments. Besides that, the idea is to maintain favorable joint angles. The use of this is the ability to maximize the impact energy that can be restored in the subsequent push off attempt. Since these are opposing tasks, the strategy offering the highest amount of stored energy while maintaining the stress level below a reasonable threshold has to be favored. The approach taken here is the combination of both by using two concurrent reflexes: The already mentioned in-flight reflex and a touchdown reflex that will be investigated in the next section. The former satisfies the need for a favorable landing position through means of position controlling the leg into a slightly bent configuration (hip angle approx. 20 deg, knee angle approx. 40 deg). The activity of this reflex (and therefore the stiffness of the joint angle controller) decreases the closer the sensed joint configuration approaches the desired one. Once the target configuration is reached within a certain threshold, the activity is kept at a level of approx. 30 %. This is done to ensure the posture remains roughly the same, even though a reasonable level of disturbance is present. The reduction of the activity has proven to be very useful at the moment of touchdown. This is because the desired amount of joint compliance can be realized in a convenient way by doing so. The stimulation is entirely withdrawn from the reflex as soon as a ground contact is detected and thus the landing can be assumed.

4.2.3.3 Touchdown Reflex

As mentioned above, the touchdown reflex is pre-stimulated at a certain point in time while the leg is still in the air. This takes place once the lateral 'overshoot' movement, caused by the push off reflex, is compensated for by the in-flight reflex. The degree of compensation is intentionally kept low in order to enable the reflex network to be immune to disturbances. This is achieved by the ability to adapt to changing environment conditions like e.g. a change of height of the ground level, caused by an obstacle. However, it is not active until the moment of impact. This

[5] Momentum applied close to a stretched out position will not significantly increase the jumping performance.

is the point in time when the activity is moved from the in-flight to the touchdown reflex. Once the ground contact is detected, it intends to gradually slow down the drop until the leg comes to a complete rest at a defined position. The former is achieved through the touchdown reflex, while the latter is managed by the squat reflex that is described in the next section. It seems pretty reasonable to consider the angular velocity at both joints as a measure for the degree of activity of this reflex as well as for the behavior's satisfaction with the current situation (i.e. target rating). The higher the angular velocity, the more counter momentum (i.e. torque) is applied to the actuated joints.

4.2.3.4 Squat Reflex

Once stimulated at the beginning of the landing phase, the squat reflex is, as already stated, responsible for controlling the leg's configuration into a defined resting position (hip angle $\approx 30\ deg$, knee angle $\approx 60\ deg$) by means of position control. By co-activating the touchdown reflex, one can be sure to reach that position with only rather low velocity. Therefore, the hardware is only exposed to a tolerable level of mechanical stress. Once the resting position is reached (lowest point of the jump trajectory), the stimulation can be set to zero and the push off reflex may be activated once more.

4.3 Results of the Simulation Process

Now that the architecture and concept of the simulation have been presented, it is time to take a closer look at the outcome. Figure 4.7 shows a series of screenshots taken during an undisturbed run of the simulated jump cycle.

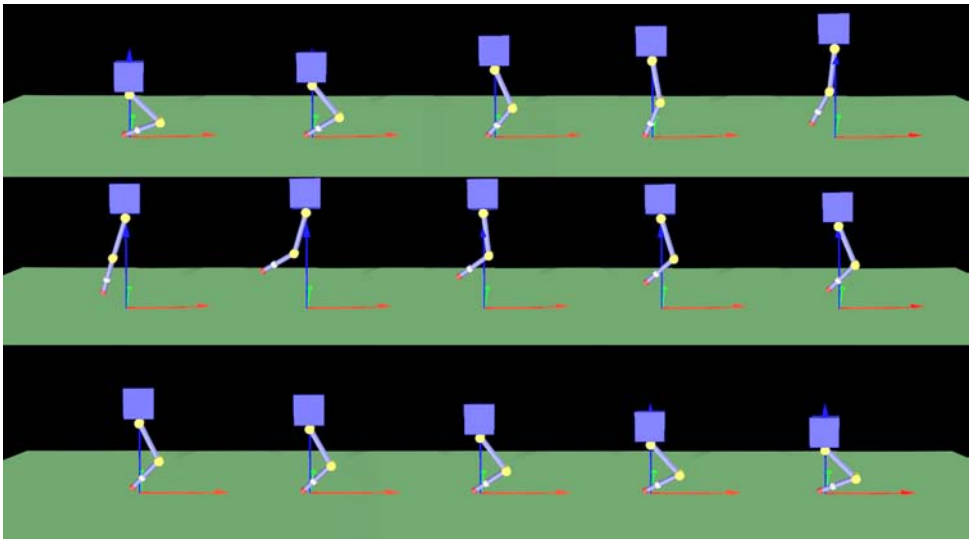


Figure 4.7: Motion capture of an undisturbed simulated cyclic jump movement

The first image on the top-left of the sequence illustrates the initial position. This squatted posture indicates the start of a cycle. In the subsequent steps (from left to right and top down) the push-off reflex is taking over control and thus causes both joints to straighten out simultaneously. This does not require precise timing since the load can be partially shifted between the actuators due to the compliance

provided by the controller. Not having to rely on a pre-scripted timing offers another important aspect besides reducing complexity during the implementation. It makes the reflex inherently robust to different kinds of disturbances like e.g. slip during this phase.

Due to a lack of torso momentum that can be employed to compensate for the leg's inertia after it loses contact to the ground, the leg initially swings backwards (top right). Now that the leg is airborne, the in-flight reflex is taking over control. It tries to servo the limb into a safe landing position as quickly as possible. For this purpose, the knee is slightly bent while the hip actuator applies full momentum to bring the leg over to the right side again. Due to the inertia and long lever arm, this takes quite a while (middle row). Shortly after, the foot point gains ground contact again and the touchdown reflex is activated. Once the impact momentum is overcome, the squat reflex adjusts the position until the initial squat is reached. Thus, the cycle can begin once more.

4.3.1 Undisturbed Cyclic Jumps

After taking a look at the qualitative results, let us now redirect the focus a little deeper into the system and investigate the quantitative outcome of the experiment. For this purpose, the data flowing inside the control architecture was recorded and is presented in this section. Figure 4.8 depicts the communication data flow in between the reflexes on the muscle group level and the feedback to the spinal cord level skill (coordinating instance) during a cycle with no external disturbance. Figure 4.9 presents the respective sensor data recorded at the same time.

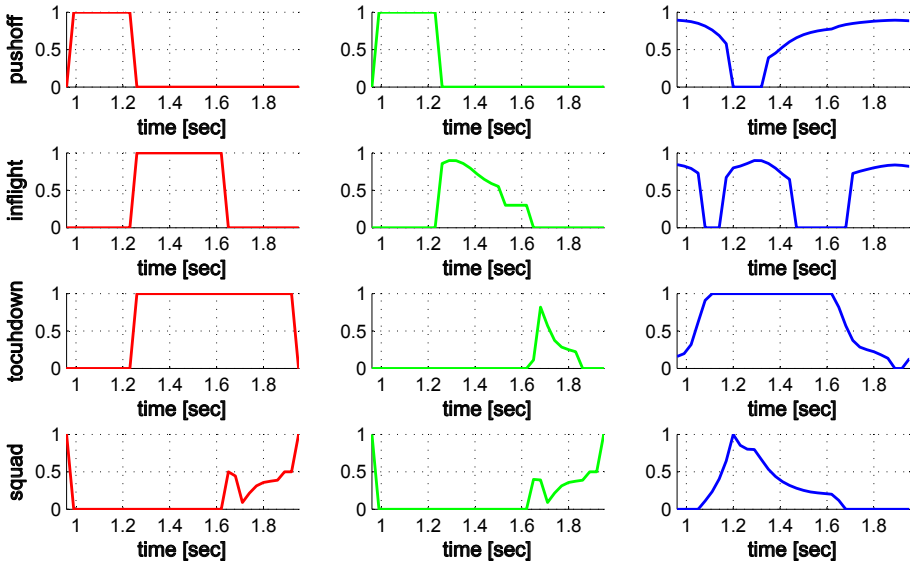


Figure 4.8: Tracking of the stimulation (red, left column), activity (green, middle column), and target rating (blue, right column) of the four reflexes controlling the cyclic jump motion

As one can see, the reflexes are arranged from top to bottom in their activity order during the jumping motion. At first the push-off behavior is active for 0.28 sec and straightens out both knee and hip joint until the lift-off occurs at simulation time 1.2 *sec*. It can be found in the sensor data as an angular inflection point. As

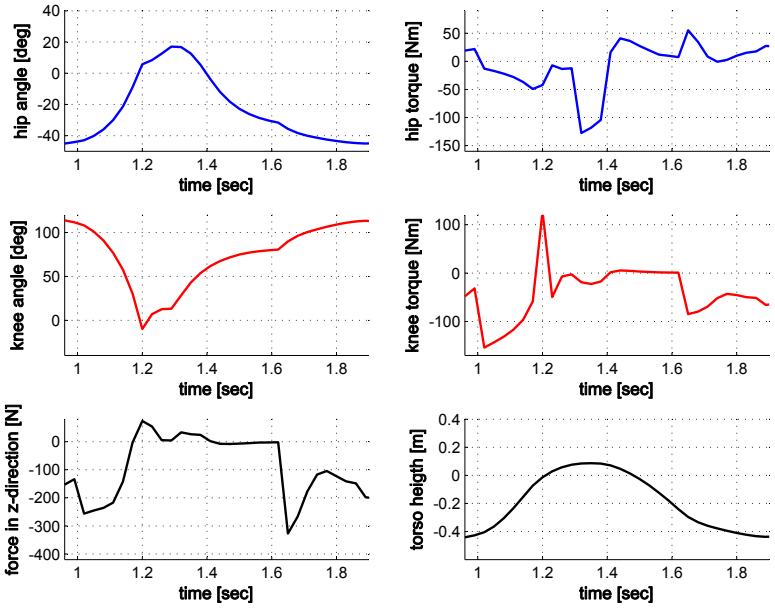


Figure 4.9: Sensor data for an undisturbed jump cycle

previously described, the full available torque is used at the knee actuator while one third of the maximum torque is applied at the hip.

During the phase when the leg has no contact to the ground[6] (1.2 - 1.63 *sec*), the in-flight reflex acquires control. As the sensor data shows, the landing position can be reached ahead of the time of impact to ensure a safe touchdown. The impact can be identified as the second inflection point in angular sensor data. The data of the load-cell underlines this assumption by showing significant change in the recorded z-force into the strongly negative direction ($< -300\ N$). As intended, the lion's share of the work during flight is done by the hip in order to perform the transition into landing position. High torques applied by the hip actuator are required to overcome inertia (1st phase) and withstand gravity (2nd phase).

After the moment of impact, the touchdown and squat reflexes take over to reduce the impact energy and maneuver the leg into its resting position. For this purpose, high torques, directed in opposition to the impact momentum are required initially. Once the angular momentum decayed past the safe threshold, the 'soft' (i.e. compliant) position controller implemented in the squat behavior takes over. As the gradual increase in the torques shows, the controller increases stiffness as it descends. This is done in order to impede oscillations in the leg that will occur at a too 'bouncy' joint configuration. The overall duration of the cycle is approximately 1 *sec*.

[6]This period is determined by a heuristic employing the load cell z-axis force. Previous experiments have shown that a force greater than -20 N reliably indicates the absence of contact.

4.4 Improving the Jumping Performance

As one can see in the previous section, the simulated performance of the robot is already quite nice when qualitative aspects are concerned. Nevertheless, energy economy becomes a very important topic if one investigates the quantitative performance measures. As Chatterjee et al. point out in [Chatterjee 02], the major aspect is to deal with the moment of collision. This is because it compromises both smooth motion and energy efficiency at the same time if disregarded. They further state that for a given actuator the overall efficiency can only be increased by reducing energy loss. This appears mainly in form of negative work of an actuator or as impact loss, to name only the two most important factors. Hence, the way to avoid impact loss is to 'eliminate' the impact mass at the contact point. This means, that the system ought to behave like a massless spring during impact. Therefore, it has to be capable of absorbing and storing the surplus energy on the one side while masking its weight by means of a steady flexion prior to the impact on the other.

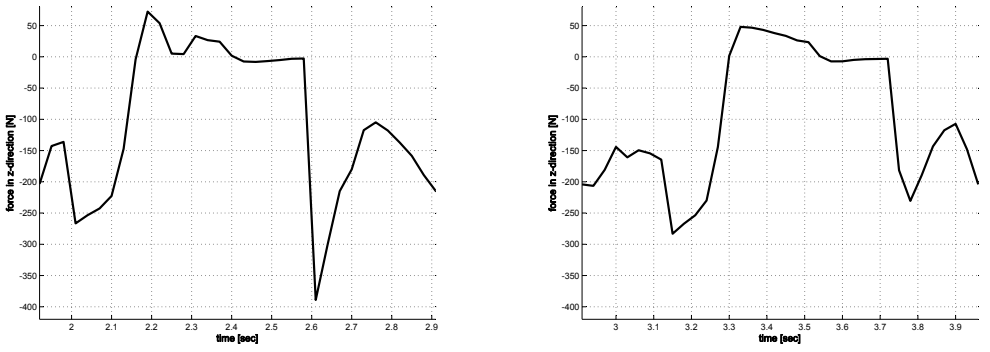
The purpose of the next two sections is to investigate the benefits and drawbacks associated with the application of springs mounted in parallel with the joints. Furthermore, a retracting reflex is introduced and analyzed.

4.4.1 Reducing Energy Loss due to Impact

As outlined above, the approach to reduce the energy loss at the moment of impact is to implement an additional fifth reflex. It is intended to initiate a steady preventive retraction motion while the leg is descending from its peak position during the flight phase. The desired effect associated with this is the adjustment of the leg's 'pretended' velocity relative to the ground at the moment of impact. This way the reflex manages to reduce the impact momentum. The effect can be illustrated best by the attempt to catch a flying tennis ball with a tennis racket and bringing it to a rest on the strings. If you keep the racket stiff during the impact, the ball will immediately bounce off. This is because the velocity of the ball relative to the racket is high. If you try to match the racket's speed to the one of the approaching ball by mimicing its motion, it will most likely come to a complete rest without bouncing. The timing is a very critical point in this context. If the flexion starts too early, the remaining angle for the transition to the resting position after the impact will be too low. Therefore, the remaining momentum cannot be compensated. The effect will be entirely lost if the flexion begins too late.

Inspired by comparable motion patterns in a human, the timing is a learning task. Therefore, several experimental runs with different timings (time of activation after leaving the ground) were performed. Of course this means that in order for the effect to unfold with the maximum influence, the peak height needs to be roughly the same. This, however, is not a major drawback here. This is because the task of repetitive jumping is performed in an environment with a pretty low amount of potential disturbances. Therefore, more or less the same height should be reached. In case the application scenario was altered, one would have to analyze the sensor data and timing of the previous cycle. For the acquired peak jumping height, the timespan best suited seems to be 240 ms after loosing ground contact. Once this time elapses, the reflex gets activated and starts to bend both knee and hip until

ground contact is established. The new behavior belongs to the class of preflexs. In this context, a preflex can be defines as a reflex that acts based on an event, that has not yet occurred but is assumed to do so at a given point in time. In this case the assumed impact is this event. The effectiveness (reduction of impact energy) can be judged by taking a look at the sensor data depicted in figure 4.10.



(a) Force without preflex **(b)** Reduced force with preflex

Figure 4.10: Plot representing the forces along the z-axis of the leg gathered using a load cell located in the lower limb

As one can see, the force occurring on the left (without preflex) at $t = 2.6$ sec is approx. 365 N and therefore more than twice as high as the one the with the preflex (175 N, right) at $t = 3.78$ sec. Hence, it is safe to say that the usage of the preflex can provide an effective means of reducing energy loss at the moment of impact. This effect is an improvement that even a significant increase in compliance during that phase cannot reach. This was proven in additional experiments that were performed.

4.4.2 Usage of Parallel Joint Springs as an Energy Storage

The introduction of the retraction reflex is a major step towards better energy efficiency. Without means necessary to store the energy, this, however, is almost useless. Thus, a second aspect has to be included into the design concept: springs mounted in parallel to the joints. Springs offer several physical advantages over entirely stiff structures. Two of those are shock absorbance and low-pass filtering of the reflected inertia. In addition, they can also be employed to prolong the impact phase. This way they reduce peak forces by spreading them over a longer period of time. Besides energy saving, this also benefits the hardware due to less wear and tear as well as less stress on the passive structures like tubes and bolts. This effect is also employed in nature as Bobbert et al. stress in [Bobbert 01]. According to their findings, the existence of elastic structures is crucial for storing energy during a vertical squat jump. This fact is underlined if one considers Hill's model. It is a model found by the English physiologist A.V. Hill describing a muscle as being composed of three elements besides the actor. A schematic view is presented in figure 4.11.

In detail those elements are a serial spring modeling the tendon, a parallel spring representing the non contractile and connective tissue within the muscle, and a

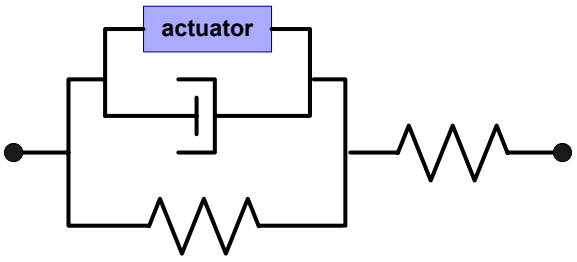


Figure 4.11: Model of a biological muscle according to physiologist A. Hill

damper emulating the behavior of the viscous elements in the muscle. The natural actuator (actin and myosin fibers) are used to contract the muscle.

At first, one might want to argue that serial springs would be better suited for the job. This is not necessarily true since their effect would be persistent. The main argument against a setup featuring elastic elements in series with the actuators is that it allows for less stiff transmission of the momentum to the ground like e.g. during push-off. If one considers the overall concept for the project, compliance is the important factor. This is because it enables the parallel spring to partially act as a serial element when the actor's compliance is high. If the current compliance state of the actor is low, the sole purpose is energy storage. Hence, the effects can be triggered when desired and suppressed when they are not. This fact represents a major advantage of such a setup.

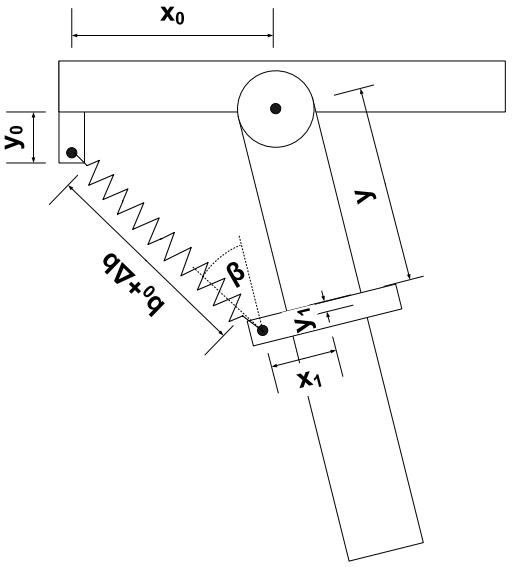


Figure 4.12: Schematics of knee joint with the parallel spring

Before the results concerning the phase coordination and sensor data will be discussed, let us first investigate the momentum introduced into the system by the springs. This extra torque can be computed as shown in equation 4.11 in connection with figure 4.12. The detailed deduction can be found in the appendix section B.2.

$$M_{spring} = \cos\beta \cdot k \cdot \sqrt{(x_0^2 + y_0^2) + (x_1^2 + y_1^2) - 2 \cdot \sqrt{x_0^2 + y_0^2} \cdot \sqrt{x_1^2 + y_1^2} \cdot y} - b_0 \cdot y \quad (4.11)$$

It basically states that besides the static variables like the mounting displacements x_i, y_i, spring parameters (constant k, initial length b_0), and the lever arm length (i.e. mounting point spacing), the momentum is solely dependent on the relative angle to the joint β.

In order to be able to evaluate the influence of the spring, it is reasonable to review the data recorded during the simulation with the new hardware setup. The parameters of the behaviors are depicted in figure 4.13 while the momenta that have to be applied by the actors during a cycle are presented in figure 4.14.

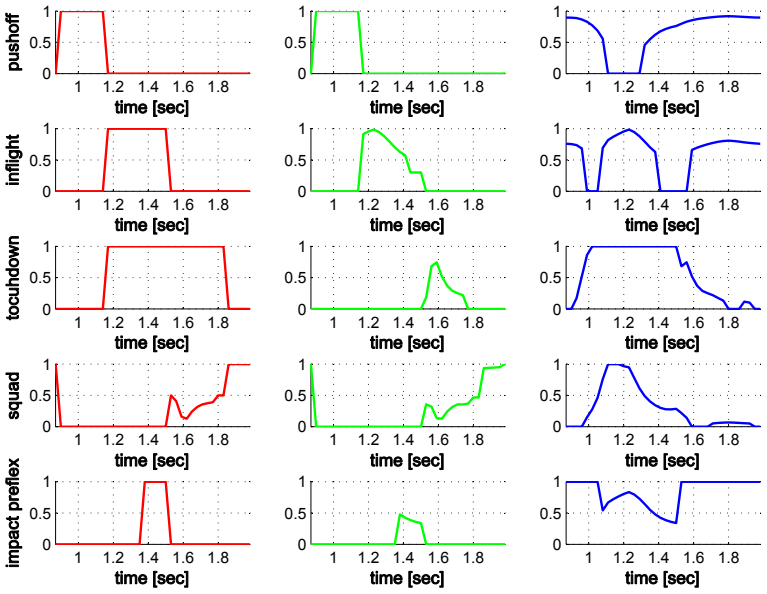
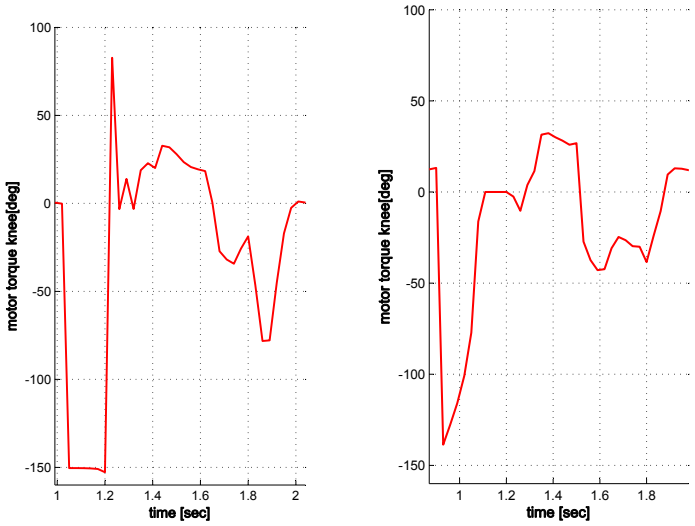


Figure 4.13: Behavior data recoded with active preflex

As one can see, the push-off phase is slightly shortened while the in-flight phase is prolonged. The former can be explained by the extra momentum stored in the spring, which can be made use of. The latter is based on the same effect resulting in a slightly increased peak height. Besides that, the reflex activity is influenced as well. During the airborne phase (in-flight, $t = 1.18 - 1.5$ sec), more activity is required in order to equal out the now opposing spring momentum. The demand for activity during touchdown ($t = 1.5 - 1.75$ sec) is reduced since the impact energy is mostly absorbed by the spring. This impression prevails when comparing the motor torques at the knee joint with springs (right) and without springs (left)that can be seen in figure 4.14.

During the push-off phase ($t = 0.8 - 1.2$ sec), the peak momentum to be applied can be reduced to approx. 127 Nm with springs instead of the former 150 Nm. Besides that, the timespan where this torque is required can be significantly reduced since only an initial 'push' is required until the spring can provide the torque by itself. This results in a significantly lower energy consumption. While in the air, the filter effect is clearly visible. It results in less abrupt movements but requires a slightly higher overall torque to maintain the spring tension. At the moment of impact, the



(a) without springs **(b)** with parallel springs

Figure 4.14: Motor torques with and without parallel springs

energy is mainly absorbed by the spring and thus a large portion of the load is no longer required from the DC motor.

Experiments performed with springs at both joints have shown that a knee spring has very beneficial effects while this concept works less well for the hip. This is elaborated from an energy point of view in section 4.5.2. It is caused by the fact that in case of the hip joint, the motor works against the spring either during push-off (inverted spring) or during flight phase (regular spring). Thus, a significant amount of energy is wasted either way. Therefore, only a knee spring will be included in the initial setup for the realization.

4.5 Assessment of the Simulation

4.5.1 Disturbance Compensation

In order to be able to fully evaluate the system performance, one must not forget to investigate the ability to adjust to external disturbances. Due to its restricted nature in respect to degrees of freedom, there are only a few sources of unpredictable effects affecting the leg. Those are the relative position of the ground in respect to the leg and the slipperiness of the surface. The former is only relevant during the phases when the system is airborne while the latter is of major importance during push-off and landing. Therefore, several experiments concerning the disturbance rejection were performed. Taking into account that the presentation of the relevant data related with the tests would be too lengthy here, the annotated figures can be found in the appendix (section B.4.1).

In the first test, the plate representing the floor in the simulation was moved up while the leg was in the air with a displacement of the approximately full peak jump height (20 *cm*, figures B.1 to B.2). It results in an impact right at the peak position representing e.g. an elevated obstacle in the landing area or a step of a staircase.

The same effect was used for the second experiment (figures B.3 and B.4) where the falling period was prolonged by lowering the plate while the leg had no contact resulting in a much heavier and later impact. Although the preflex is solely time triggered, the network was able to adapt to both cases while continuing to pursue the jumping task. It has to be pointed out that the leg is capable of compensating disturbances that are a lot higher than the ones presented here. This is because of the very robust behavior based control architecture. During a series of tests, the ground level variation could be altered during an arbitrary jump phase up to more than the full extension of the leg without compromising the jump task. However 20 *cm* were selected since they represent a more typical magnitude considering the overall dimensions.

The last test focuses on the effect of slip during the push-off phase. For this purpose the plate is quickly moved in lateral direction shortly before the leg is about to lift off. The data can be found in figures B.5 and respectively B.6 for the sensor data.

4.5.2 Evaluation of the Energy Efficiency

The second major concern besides distortion rejection that deserves an in-depth analysis is the energy efficiency of the system. For this purpose the torques applied by both motors were recorded for the duration of a simulated jump cycle. The data is depicted in figure 4.15. The hip torque is plotted in blue on the left while the applied knee momentum is drawn in red on the right.

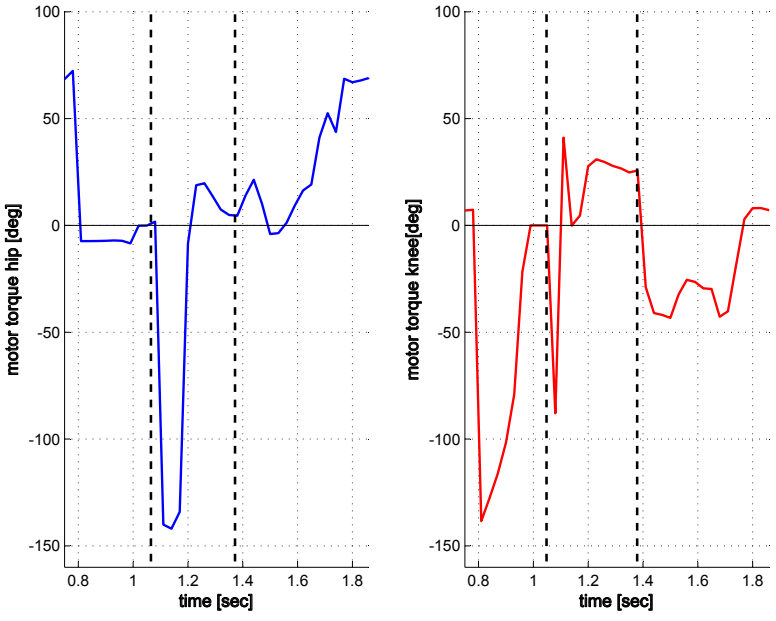


Figure 4.15: Torque of motors applied to the hip (left, blue) and knee (right, red) joint.

It is obvious that during the push-off phase ($0.8 - 1.05$ *sec*) the main share of the work is performed by the knee actuator. This holds true until the torque is withdrawn shortly before full extension at $t = 1.0$ *sec*. The idea behind this is to

reduce the inertia momentum while in the air[7] and save energy at the same time. This effect is obvious if one employs basic trigonometry on the triangle of forces at either joint: The vertical contribution decreases while the lateral one increases by the same amount.

After lift-off, the main concern is to reach a safe landing position as soon as possible. This in necessary in order to be prepared for the impact. Thus, the initial torques applied during this phase ($t = 1.06 - 1.38\ sec$) are rather high for the hip actuator. The unequal distribution of the load can be explained by the fact that both inertia and gravity have to be overcome to retract the leg. Due to the kinematic setup, this can only be done by the hip. The increase in knee momentum is less significant. This is because it only has to move the mass of the lower limb while the hip is burdened with both limb segments as well as the weight of the knee actuator and gearbox. Once an acceptable landing position is reached at $t = 1.2\ sec$, the preflex becomes active and causes a steady torque to be applied until the impact at $t = 1.38\ sec$.

The initial momenta for the landing period increase to limit the angular velocity at both joints. Subsequently the position controller takes over and servos the leg into its resting position with increasing stiffness.

In order to be able to determine the energy efficiency on a quantitative level and thus get a more objective point of view, the energy applied was computed for each cycle for four different setups by integrating the applied momentum over time.

Setup without springs:

Jump phase	Duration [sec]	Energy hip [J]	Energy knee [J]
Push-off	0.3	152.0	1006.0
Airborne	0.35	487.8	239.9
Landing	0.38	269.0	406.4
Σ	1.03	908.5	1652.2

Setup with knee and hip spring:

Jump phase	Duration [sec]	Energy hip [J]	Energy knee [J]
Push-off	0.33	477.1	616.1
Airborne	0.36	649.3	215.9
Landing	0.33	226.1	351.6
Σ	1.02	1322.5	1183.6

Setup featuring an inverted hip with the regular knee spring:

[7] Due to the absence of a DOF for the lower trunk, the inertia has to be compensated for by the hip actuator.

Jump phase	Duration $[sec]$	Energy hip $[J]$	Energy knee $[J]$
Push-off	0.3	435.6	621.0
Airborne	0.35	517.9	333.1
Landing	0.62	1309.1	454.0
Σ	1.29	2262.6	1408.1

Setup including the knee spring alone:

Jump phase	Duration $[sec]$	Power hip $[J]$	Energy knee $[J]$
Push-off	0.29	192.8	598.8
Airborne	0.36	530.8	396.9
Landing	0.4	474.0	356.3
Σ	1.05	1197.6	1352.0

As one can clearly see, the configuration with an inverted hip spring is not an option. This is mainly because such a spring setup dramatically increases the energy demand rather than decreasing it. Based on the results of this experiment, the setup featuring just the knee spring without a hip spring was chosen amongst the remaining three options for several reasons:

- The combined energy consumption is amongst the lowest compared to the others.
- As opposed to the setup without springs, the workload is almost evenly distributed between hip and knee actor. This reduces wear and tear and allows for less robust actuators and thus helps to reduce the weight.
- The average torque applied by the knee during push-off is almost halved due to the energy stored in the spring.
- It offers about the same energy efficiency as the setup with both springs while reducing the weight and thus stress put on the structures by e.g. impact and inertia. Thus, one can state that the hip spring has no significant advantage over a springless setup. Furthermore, it increases the torque demand during the most volatile phase of the sequence, i.e. while in the air and thus increases the danger of not being able to reach a safe position prior to impact.

Hence, the further experiments are performed using setup four since its properties are superior to the others.

Now that the issue of spring configuration is settled, let us continue to assess the system's performance compared to humans. The human ability to perform energy efficient jumps has not even remotely been matched by an engineered device. Therefore, the human represents the ultimate benchmark for the system's performance. The data of human squat jumps considered in the following paragraph can be found in [Fukashiro 05].

The test subjects for the experiments are male athletes. In order to produce a fair basis for comparison, the power to weight ratio (PWR) has to be adjusted first. The human PWR of combined approximately 2352 W applied by three powered joints (hip, knee, ankle) is slightly less than the system's 2428 W in two actors. In a human, the share of knee and ankle joint momentum is almost equal. However, they are applied at different phases during lift-off and landing in order to sustain a roughly equal force. The particular timing is the key element in order to achieve maximum efficiency.

Besides the PWR, the peak torques are of major importance. In a human the applicable torques recorded[8] during vertical squat jump are for the hip 112.8 $\pm$ 40.5 Nm, for the knee 118.7 $\pm$ 34.7, and for the ankle 96.1 $\pm$ 24.4. Based on the fact that the subjects were not allowed to use arms, the data acquired may serve as a basis for comparison although ankle, torso, and head momentum cannot be reproduced in the experiments on the system. Nevertheless, the maximum torques applied by the DC motors are roughly equal to the ones found in humans. The highest hip torque (138 Nm) is reached during the airborne phase while the knee torque of 145 Nm is applied during push-off.

In order to produce directly comparable results, one has to eliminate all sources of actuation that cannot be reproduced on the robot. For this purpose the contributions to the take-off speed in a human that can be found in [Luhtanen 78] were used to get rid of all sources of actuation that cannot be employed during the simulation. Those are 22 % planta[9] flexion, 10 % trunk extenison, 10 % arm swing, and 2 % head swing. Thus only the 56 % of the remaining leg actuation is used as benchmark.

Due to the fact that the velocity is squared to determine jump height[10], the lift-off velocity was reduced to the results of the leg efforts alone. The result of a scaled human performance would be an approximate peak jump height of 15.87 cm. Compared to the recorded 12.6 cm by the system, this results in an overall effectiveness of approx. 79.4 %. This is quite remarkable and higher than initially expected. The graphical interpretation of this number is that if the same amount of energy is applied to a comparable kinematic setup, the robotic leg will only reach around 80 % of the human peak jump height. In order to be able to acquire an even better result, one has to think about including an ankle joint for the realization. Since an actively actuated ankle would be by far too heavy due to the long level and thus high momentum, passive actuation by a spring has to be taken into account. Besides the extra stored energy available for the push-off attempt, a series elastic element would also be in favor of the controller performance. The combination of an actual elastic element and a 'software spring' realized through the compliant controller can be employed in case of a distortion like e.g. impact. It would allow the control architecture to react to the event with a certain tolerable delay. This time period can be used to determine the reasonable response. Thus, overall robustness can be dramatically increased.

[8] considering the standard deviation

[9] *lat.* sole of foot

[10] $h = \frac{v^2}{g}$

4.6 Comparison to a Human Squat-Jump Trajectory

The intent of this section is to compare the human squat-jump motion to the one realized during this project from a trajectory point of view. Before we can do so, however, one has to understand where the artificial trajectory originates from. In absence of a globally defined trajectory in the simulation, it can be understood as a result of the reflex cooperation as well as the system's inherent natural dynamics.

The data for the human squat jump was acquired using a state-of-the-art digital camera with a high speed shutter program. Dependent on the light situation, up to approx. 20 pictures per second were taken. In order to reduce the error margins during the analysis of the data, markers were positioned at the subject's joints. The angles relative to the markers, i.e. relative segment angles, were determined using a standard PC image editing program. The results are depicted in figures 4.16 and 4.17. The simulation data is marked in blue for the hip and red for the knee joint. The human data is indicated by the markers. A projected trajectory is determined by means of cubic Hermit interpolation between those points.

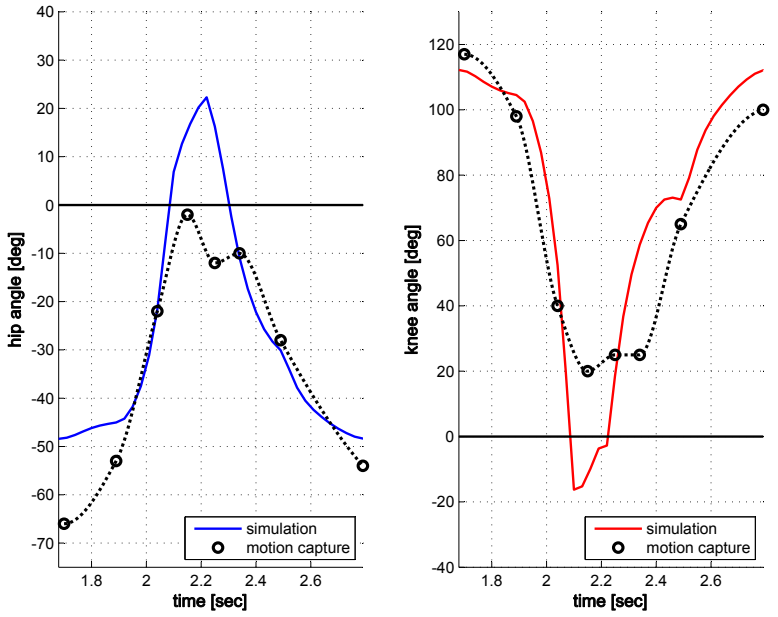


Figure 4.16: Data gathered during motion capture of a male subject

The superficial result of these considerations is that the recorded trajectory during simulation and motion capture are in large parts identical. Considering the fact that the trajectory was not explicitly designed to be human-like, this is a rather good indicator that the natural dynamics of the system can unfold almost as freely as intended. The most significant deviation can be found after the lift-off has occurred. This is because the robot is unable to compensate the leg's inertia like its human counterpart. Thus, the angle 'overshoots' the straight position by about 20 degrees in both joints. Besides that, one can state that the initial and end position of the

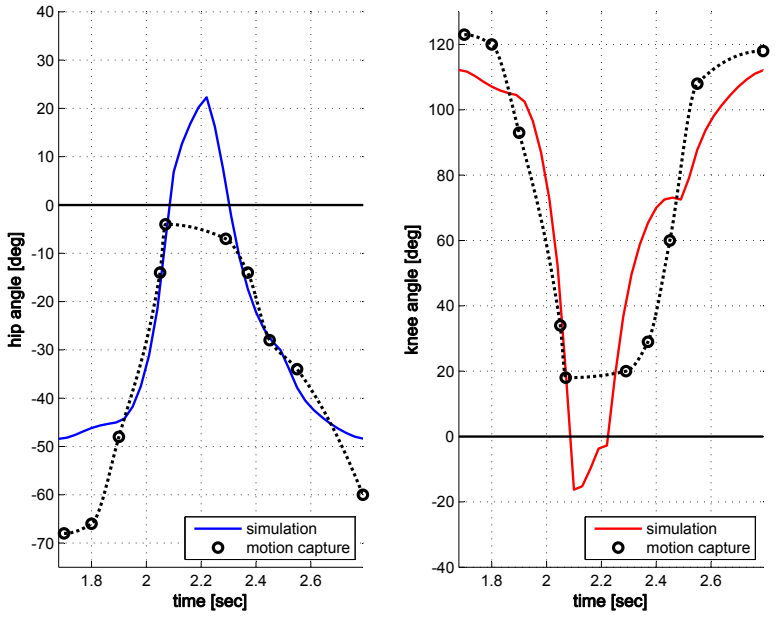


Figure 4.17: Motion capture of a second male subject

subjects were deeper (resulting in a larger angle of approx. 10 *deg*). This position cannot be reached by the robot since it is unable to shift its hip in lateral direction like a human would do by adjusting its torso. Besides the deviations, it is noticeable that a human shows a behavior comparable to the preflex motion. By bending the knee, and thus retracting the leg, the shock momentum transmitted by the osseous structure is minimized. This can be achieved by using the elastic muscle structure to reduce the impact. The most significant part of the energy is compensated by the foot mechanism. To be more precise the Achilles tendon is storing a large portion of the impact momentum. Therefore, a more significant retraction is not needed.

5. Realization

The task of this chapter is to present the construction of the leg prototype and the realization of the control architecture. In addition, several experiments will be discussed.

5.1 Mechanical Design of the Leg

The leg prototype constructed for the experiments originates from the works of Luksch et al. [Luksch 07]. The given setup is intended to model the basic functionality of a human leg. Thus, the mechanical construction is comprised of a femur segment[1] and the lower thigh section. In the biological role model this section is subdivided into the tibia-[2] and fibula section[3]. The anatomy of a human leg can be seen in figure 5.1.

Since no twist in the tubular structure is obtainable, the lower thigh can be approximated by the tibia alone. The approach taken in this thesis is slightly different from the one of the previous works on the prototype leg. Therefore, some components had to be reconsidered. The general tube structure (imitating the bones), as well as the actuation concept remained unchanged. This is because the selected disc runner motor (Heinzmann SL 120-2NFB) incorporates all major features that are required for a highly dynamic motion pattern. The major characteristic of this class of motors is a very low actuator inertia. This ensures fast reaction times. The ability to deal with high current for a short time underlines the high variability concerning the torque. Another central aspect of the actuation concept are the gear boxes. Their reduction ratio was kept at 32:1. This low ratio is important in order to achieve a non-retardant behavior concerning the motion restriction. Thus, the already present gears[4] were maintained.

Other parts had to undergo a redesign process since they were no longer suited for the new application. Amongst these is the pneumatic knee spring. Since the practical

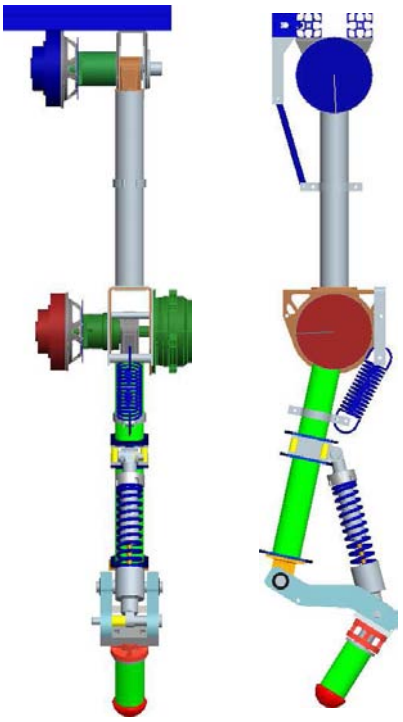
[1] (lat.) femur: thighbone
[2] (lat.) tibia: shinbone
[3] (lat.) tibia: calf bone
[4] hip joint: Neugart PLE-60-32, knee joint: Neugart PLE-64-32

Figure 5.1: Anatomy of the human leg. On the right limb the muscles are depicted while on the left the illustration is reduced to the bone structure and the ligaments. The illustration is taken from 'The Visible Body' project (www.visivlebody.com)

tests of Luksch et al. revealed that it cannot live up to its theoretic potential in reality, it had to be replaced. As pointed out in the previous section, springs seem to be essential for robust and efficient locomotion. This is for multiple reasons: They provide energy storage, high mechanical power, and are capable of overcoming bandwidth limitations of traditional actors.

In traditional actuation (utilizing DC motors and gears) the highest energy loss during a jump cycle occurs at the moment of impact. By using springs this energy can be preserved if the natural frequency of the mechanical setup is about equal to the gait cycle. This spring like behavior can be found in all kinds of running animals from insects to large mammals. Here the center of mass performs a motion like a bouncing ball. Besides energy conservation, springs also help to reduce the weight of the actuators. This is because DC actors in comparison to springs require approximately 30 times the mass for an equal power output according to [Hurst 08]. The usage of springs can be motivated by the existence of its biological counterparts, the tendons. As seen in the variety of tendon structures in nature, dimensioning of such components is a very important factor. If the spring is too stiff, it is capable of storing high amounts of energy but rather suppresses the natural dynamics. If it is too soft, the energy storage capability will be dramatically reduced. The final setup featuring parallel springs can be found in figure 5.2.

As one can see, the setup was extended by adding a rotational DOF (mechanical foot) at the end of the kinematic chain that can act as the ankle joint. Besides this foot setup, two other assemblies will be tested later on. Since an actively powered ankle would require very high momenta because of the unpleasant lever ratio, it is



(a) Side view **(b)** Front view

Figure 5.2: Technical drawings of the leg prototype taken from the CAD software.

reasonable to use a spring instead. This is because of the (already mentioned) very high power to weight ratio. Hence, passive compliance with a low overall weight can be realized at the same time. If one considers the selected setup, it becomes obvious that the spring acts as a series elastic element. This offers several advantages for the task of jumping: It allows to reduce the disturbance propagated into the system because the foot shows a low-pass filter-like behavior.

Besides the already mentioned features, the concept of gravitationally decoupled actuation (GDA) and coupled drives (CD) as presented in [Pfeiffer 07] can be utilized. The latter is of special importance to reduce the setup weight. Making use of the selective compliance (provided by the joint controllers), one is able to distribute the load amongst all actors in the kinematic chain. This way the requirements concerning the acquirable peak output for each actuator can be lowered. Thus, smaller motors can be used. In conclusion, one can state that a combination of DC motors, parallel springs, and a compliant joint control seems to be ideally suited for the task of performing repetitive jump motions employing natural dynamics. This is because the most prominent demands for such a task can be met: High power density, energy storage, and a dynamic energy profile. At the same time a reasonable weight and complexity level can be maintained.

5.2 Transfer to the Embedded Platform

In order to be able to utilize the control concept as presented in chapter 3 and 4, it had to be implemented on an embedded platform. For this purpose the controller had

to be ported onto a DSP circuit board that is used to control the power supply. The control architecture on the other side was already implemented on the PC. However, it had to be adapted to be able to communicate with the DSP. Therefore, the physics simulation was replaced by a DSP remote part. This part is (in combination with the already established framework) capable of managing the information flow between the PC and DSP environment via a CAN bus interface.

5.2.1 Distributed Control Concept

As mentioned before distribution is a logical consequence of closely coupled, delay critical processes. Hence, a concept had to be established that accounts for this. In case of this project, high level skills, intermediate level reflexes, and low level joint controllers had to be assigned to either PC or DSP. In order to do so the best way possible, the problem domain-specific knowledge has to be considered. An illustration can be found in figure 5.3.

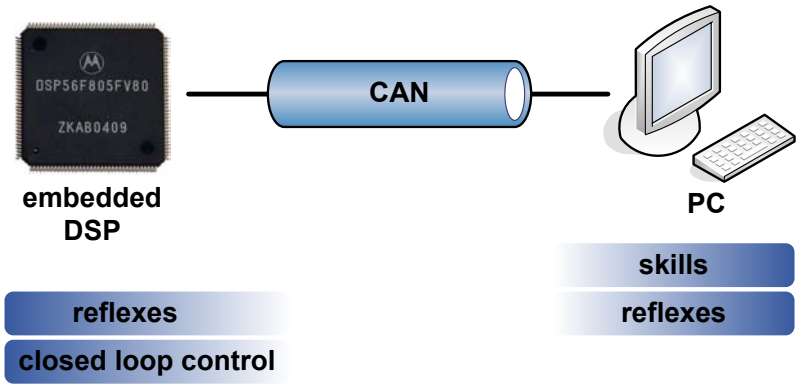


Figure 5.3: Concept concerning distributed control realized in this project

The PC is designed to host all less time critical components. This is because the employed standard CAN bus does not offer real-time properties. For the initial test configuration of the system, all reflexes as well as the skills were realized on the PC side. The DSP's sole assignment for now is to accommodate the joint controllers since they are the most time critical components of the architecture. Once the approach has proven to be capable of performing a controlled cyclic jump, the next step will be to port some of the time critical reflexes (e.g. touchdown and in-flight reflex) to the DSP and investigate the benefits for the overall process. This can be done easily since the reflexes themselves are of very basic structure. Therefore they can be ported into very few lines of code.

5.2.2 Implementation of the Control Algorithm

Since the simulation was entirely performed using MatLab Simulink, the controller had to be reimplemented in the programming language C in order to be used on the DSP. Besides this, a few adaptations to the new platform had to be performed. Other components were added as the controller was tested to improve its performance. The control algorithm for the position, speed, and current controller can be found in the appendix (section B.3).

The two most important changes due to the porting to the new environment are the need for heuristics concerning the current sign as well as a low-pass FIR filter[5] for the speed controller sensor input. On the DSP the current is sampled behind the H-bridge circuit that is used to power the motor. Therefore, the direction of the current flow, as opposed to the magnitude, cannot be directly measured. Thus, it becomes necessary to introduce heuristics in order to reconstruct the direction based on the information accessible. During the test phase, the most basic solution turned out to deliver the best performance. It even turned out to be better than the fancy current mapping approaches that were pursued. Thus, the current sign is determined based on the sign of the PWM. The assumption made is therefore, that if the PWM output is positive, so is the current.

The second adjustment that had to be made is the FIR filter. It was introduced because initial experiments had proven that the controller reaction to the noisy speed information (gathered by the joint encoders) turned out to be too erratic. This is because the controller was set to act quite aggressive. This setup is required in order to be fast. To compensate for this trade-off, the data was artificially smoothed using a first order FIR filter. Doing so allowed for receiving a low controller delay while ensuring sufficient stability.

The second group of adjustments contains several minor additional components that were introduced in order to optimize the controller performance. Among these are limiters and an anti-windup mechanism. The former was introduced to tune the controller to the desired dynamic behavior. For this purpose the output of e.g. the torque controller was bound within a threshold in order to reduce the impact of the overall controller output. The anti-windup mechanism, introduced in the integral-portion of the speed controller, has proven to be of great use, too. This is because the difference in response delay between the torque and speed controller is so severe (due to the cascaded structure), that the speed controller will run into a integrator windup if this is not prevented. Initial experiments have shown that the new setup is superior over the old one because of the less lazy reaction to change in the plant or the reference input.

5.3 Experiments

Before the motors are mounted on the leg setup, an experiment concerning the controller reaction to disturbance is performed. For this purpose both motors are connected by single shaft. Thus, they are turning in opposite directions. Due to the stiff mechanical connection one actuator can be controlled while the second one acts as the defined load. Before the actual controller parameters are tested, the PWM limit had to be determined. The upper limit was set at the end of the linear region in order to ensure a proper controller behavior. After this is dealt with, the controller parameters are determined.

Since most parameters are mutually dependent, a structured approach had to be followed. Therefore, one has to start by setting the torque parameters and work one's way up to the position controller. At first, the proportional parameters (DPG, APG) are set in accordance to the observed reaction to a step-like change in the

[5]FIR = Finite Impulse Response

reference signal. The gains are increased until the first signs of instability occurred. The last setting that guaranteed stability while allowing for the fastest reaction is selected. After that, the integral portion parameter is set. It is of special importance to make the integral part strong enough to quickly eliminate static error but not too strong. This would allow it to overpower the proportional portion and therefore generate an undesired controller behavior.

Once all parameters for the controller are set, the dynamic characteristics have to be determined. Thus, the limiters and sensor filter coefficients are investigated. The limiters are means to reduce the dynamic characteristics. This is required due to the cascaded structure. Based on this, the outer controller is only allowed to change its output (that serves as the reference input for the controller on the lower level) at a rate that does not cause the next controller to become unstable. This is done by limiting the maximum step size per cycle time. The filter coefficients are even more important when the stability is concerned. If it is set too high (narrow low-pass filter) the controller will work really smooth but will be very slow. If it is set too low, however, the controller will react to noise too heavily and therefore become unstable. Thus, the setting of these parameters turns out to be a balancing act between speed, accuracy, and stability.

The speed and the position controller are set the same way. This process results in more than 15 parameters to be set for each joint controller. What makes this task even more difficult, is the fact that stability and performance have to be guaranteed for any given combination of the two iota settings, that have fundamental influence on the controller behavior.

5.3.1 Controller Performance Assessment

After all parameters were set and conscientiously tested, experiments are performed to determine the individual controller performance. The first controller investigated is the torque/current controller. The result of the experiment is presented in figure 5.4. The reference input is depicted in red while the system output is marked in blue.

The performance is analyzed based on a step-like change in the reference signal of approx 7.5 A current. This scenario is considered to be typical for the given application. As one can see, a rise time of approx. 9 ms can be observed. Compared to the 4 ms that were achieved during the simulation, this seems to be a rather poor result. However, it becomes rather obvious, if one considers the difference between the simulated and the real controller environment. The presence of noise in the sensor data, as well as delay and other undesired effects that were not modeled in the simulation, causes the parameter setup to be very different. This also shows in the quite significant overshoot of 28%. Since it is not critical, this has to be accepted as a trade-off for a fast and at the same time stable controller. In order to reduce the overshoot, the gains or filter coefficients would have to be lowered significantly. This would cause the controller to become slower by at least a magnitude. This would make it no longer sufficient for the given highly dynamic application. Hence, the found parameters remain unchanged. In order to ensure stability, the change in reference signal for this controller is limited to $\pm$ 0.8 A per cycle at a cycle time of 1 ms. This value has proven to be more than sufficient for the task at hand.

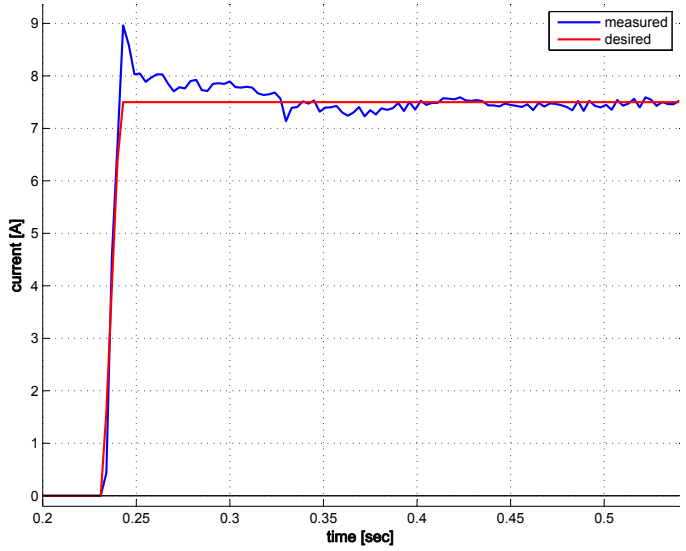


Figure 5.4: Performance of the implemented torque controller in respect to the measured response (*blue*) to a instant change in the command signal (*red*). The data is based on a sample rate of $333\frac{1}{3}$Hz

The next controller to be investigated is the speed controller. Again a step-like change in the input is used to stimulate the system. The recorded output can be found in figure 5.5.

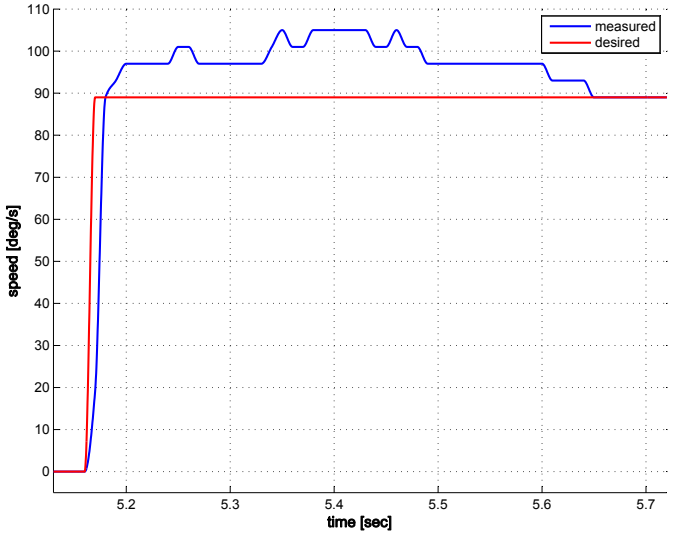


Figure 5.5: Response of the speed controller to a step-like change in the reference signal. The recorded velocity is denoted in blue while the commanded signal is marked in red. The sample rate used is 100 Hz

The step size is set to 90 deg/s. Although the speed controller is stacked on top of the torque controller (and thus makes use of it to access the plant), a rise time of 29 ms could be acquired during the experiment. This seems rather odd because this is only half of the time that was needed during the simulation (rise time 56 ms). However it can be explained by the more aggressive controller setup. The trade-off for this is again the increased overshoot. This is also clearly visible in

the recorded results in the figure. An increase in the integral gain is not advisable since this would limit the dynamic range significantly. Nevertheless, the overshoot is acceptable if one considers that the speed controller solely serves as a slave controller. Therefore the position controller can take measures to compensate it. Hence, the overall performance can be assumed to be as intended.

The last controller to be assessed is the position controller. Once more, a step-like change in input of approx. 44 *deg* is utilized. The system's response is shown in figure 5.6.

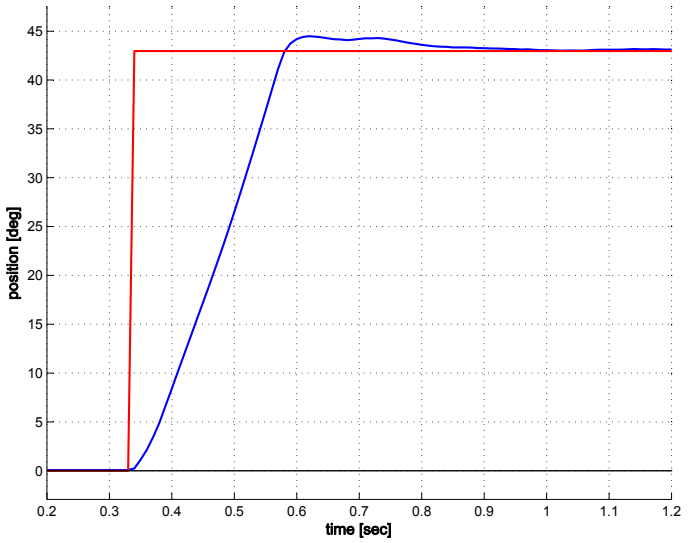


Figure 5.6: Recorded reaction of the position controller to a change in commanded input of approx. 44 deg. The reference signal is marked in red while the measured position is plotted in blue

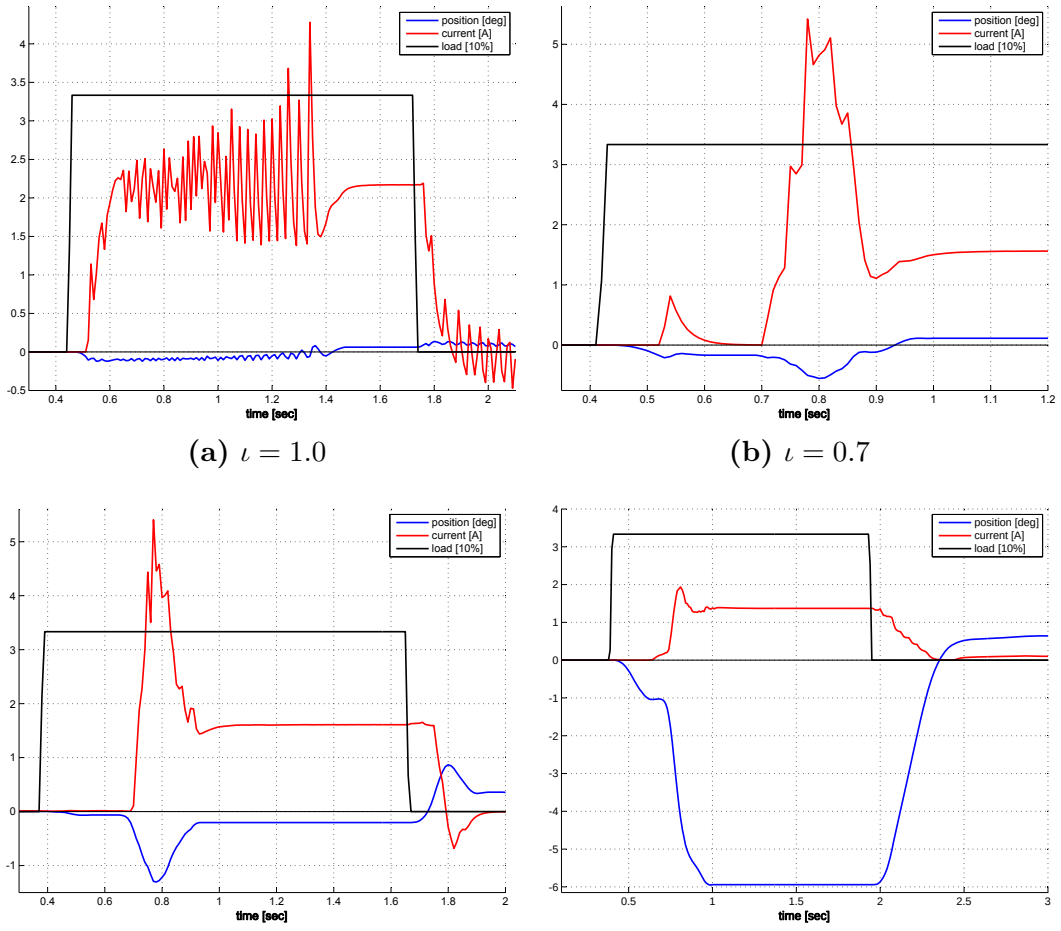
As one can see the top-most controller is also the slowest. This is because it has to make use of cascaded structure below itself. Thus, the summed delay of both controllers limits the position controller's performance. The rise time of approx. 180 *ms* leads to a stable signal after 240 *ms*. The real result is only a little bit slower than the outcome of the simulation process. The overshoot of approx 3.5 % is a good indicator for a fast setup. The dimension of the integral portion does also turn out to be reasonable. This can be seen in the steady output delay and an error level below 0.1 %.

5.3.2 Compliant Capabilities

5.3.2.1 Static Compliance

Now that each individual controller is assessed, it is time to finish off the paragraph by investigating the impact of the compliance on the performance. For this purpose, four different settings of the position compliance parameter are tested. The results can be seen in figure 5.7.

In all figures the blue graph indicates the position given in degree, the red represents the current in ampere, and the black line indicates the load applied by the second motor. Compliance is expressed in this context solely by ι. A value of 1 indicates



(a) $\iota = 1.0$ **(b)** $\iota = 0.7$

(c) $\iota = 0.4$ **(d)** $\iota = 0.1$

Figure 5.7: Performance of a compliant position controller based on the stiffness weight $\iota \in [0, 1]$. The red line represents the current measured in Ampere, the blue one stands for the current position denoted in degree while the black graph indicates the load applied by the second motor in percent

an entire stiff behavior while a value close to 0 will result in entirely compliant characteristics. In all cases the controller parameters except for ι remained unchanged. A steady load of 33 % of the peak momentum was applied in either case as a source of disturbance.

The first setting investigated is depicted on the top left (figure 5.7(a)) with a setting of $\iota = 1$. As one can see the controller's tolerance for the position error is quite low (approx. $\pm$ 0.1 *deg*). Thus, a very fast and heavy response is initiated. This leads to heavy initial oscillations until a steady state is reached. The maximum observable position deviation can be estimated with 0.2 *deg*. This indicates that the controller is borderline stable. Due to the fact that such a behavior will put an enormous stress on the actuators and the mechanical structures, it seems not to be well suited to be used in this class of applications.

The second setup that is investigated is a setting of $\iota = 0.7$. The data is depicted in figure 5.7(b). Here the position error tolerance ($\pm$ 0.2 *deg*) is slightly higher and the controller reaction to the distortion is less severe but still harsh. The maximum observed position error is approx. 0.5 *deg*. The most prominent difference to the previous setting is the absence of significant oscillations.

The next experiment was performed after setting $\iota = 0.4$. The results are presented in figure 5.7(c). As can be seen, the position error tolerance increases once more to approx. $\pm$ 0.3 *deg*. The counter momentum applied by the controller decreases even further. This allows for a maximum error of 1.2 *deg* to occur. The behavior found here underlines the advantages of compliance in the presence of disturbances. The controller allows for a reasonable amount of error before introducing a rather soft but steady counter momentum.

The last case considered is a setting of $\iota = 0.1$. As expected, the controller hardly shows a response. Therefore, the position error reaches a level of approx. 6 *deg* before the disturbance momentum and the counter momentum applied by the controller equal each other out. The overall position error tolerance can be estimated as $\pm$ 1 *deg*.

In conclusion, one can state that with increasing ι the dead band broadens. This is because the tolerance to deviation increases due to less occurrent controller input error[6]. Another aspect of this is that the more compliant the controller is set to be, the slower the initial position is reacquired after the distortion. Further experiments involving more iota settings have shown that the optimal compliance parameter setup ranges between 0.2 and 0.7. Thus, these settings will be used by limiting the maximum ι propagated by the hardware abstraction layer to the DSP.

5.3.2.2 Compliant Position Control

Besides the compliant properties in case of external distortion, the performance concerning the position control with limited stiffness is still not investigated. In order to allow for free and undisturbed movement, the slider was used to keep the leg floating in midair. Subsequently the knee joint is servoed from a straight configuration (lower limb perpendicular to the ground) into a joint angle of 45 *deg*.

[6]This is because the ι parameter is multiplied with the error. Thus, the error visible to the controller gets less significant with decreasing stiffness

This experiments are repeated for multiple stiffness settings. The results concerning the duration and the acquired resting angle can be found in table 5.1.

Stiffness $[\iota]$	**Resting Angle** $[deg]$	**Duration** $[s]$
0.20	31.19	0.75
0.30	45.13	0.66
0.40	44.7	0.54
0.55	44.5	0.45
0.60	44.79	0.42
0.65	44.67	0.42
0.70	44.72	0.41
0.75	44.65	0.39

Table 5.1: Overview of the controller performance for various compliance settings

It can be seen that even for a rather low stiffness ($\iota \leq 0.3$) the target angle is reached reliably. The remaining position error for high stiffness can be assumed to be below 0.5 *deg*. Furthermore, it is noticeable that the timespan required to move the limb into its resting position decreases dramatically for higher stiffness. Thus, only a little over half of the duration is needed for a setting featuring low compliance ($\iota = 0.75$) compared to a very compliant one ($\iota = 0.2$). In conclusion, one can state that the observed positional precision and time delay are more than sufficient for the task if the most critical phase (in-flight) is assumed as a benchmark.

5.3.2.3 External Distortion

For the next experiment the slider is lowered again in order to allow the leg to bounce up and down freely while maintaining steady contact to the ground. The hip joint is entirely relaxed while the knee was kept at a position of approx. 40 *deg*. A repetitive pattern of momentum is applied to the slider in negative z-direction at different compliance settings. This allows for determining the controller performance in a more realistic scenario. All major system parameters are presented in figures 5.8 through 5.10. Each experiment record includes the hip and knee angle as well as the controller directed current and the load cell data.

Initially a very high stiffness setting of $\iota = 0.9$ is assessed. The result of this is that the leg could not be significantly moved. Thus, only lower stiffness settings are presented in the following paragraphs. Figure 5.8 illustrates the system behavior at a setting of $\iota = 0.7$.

Due to the high stiffness, only a low angular displacement of approx. 10 *deg* can be acquired at the knee. A large portion of the disturbance is compensated by the applied motor torque. This can be seen in the data if the motor current is investigated. Large amplitudes greater than 15 A can be seen. Thus, the major share of the shocks is not propagated to the load cell. This is obvious since besides the general noise, no peaks can be identified in the z-direction force. At this setting the joint configuration seems to be too stiff to transmit load to the adjoining powered joints. Therefore, higher compliance is needed when the concept of coupled drives is to be applied.

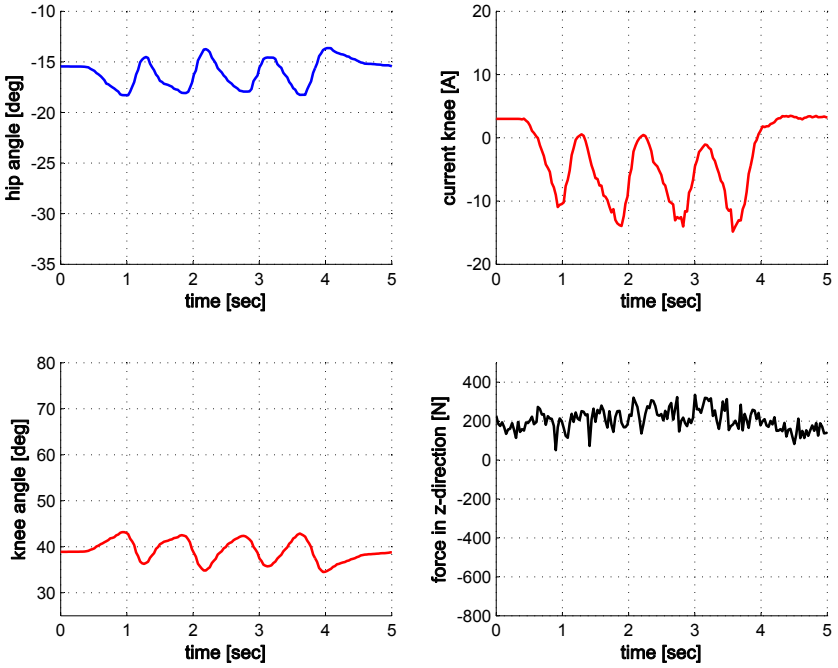


Figure 5.8: Data recorded while the leg setup was exposed to external distortion with a compliance setting of $\iota = 0.7$

For the next experiment, the joint stiffness was reduced to a value of $\iota = 0.5$. The results are presented in figure 5.9.

With the new setting and an approximately equal momentum at the slider, higher angular displacement at the knee joint (approx. 15 *deg*) can be observed. Furthermore, the motor currents are lowered to an amplitude of less than 12 A. This represents a more compliant characteristic of the joint. This is also visible when the load cell data is considered. High force peaks with a dynamic range of approx. 900 N can be seen. Moreover, they seem to be closely coupled with the current peaks. This leads to the conclusion that the momentum is now propagated along the tube structure to the adjoining joints. This setup seems to be a good compromise between stiffness and position accuracy.

The last experiment of this series illustrates the behavior of a very compliant joint. Thus, the stiffness parameter is set to $\iota = 0.3$. The results are shown in figure 5.10.

Using this configuration, the leg shows very bouncy characteristics. This manifests in a large angular displacement at the knee of more than 40 *deg*. The occurring shocks cause only very little controller reaction in form of motor current. Thus, the major share of the distortion is propagated along the mechanical structure and passed to the load cell. This results is an increased duration and a respectively lower amplitude of approx. 500 N. This is most likely the case because the parallel spring at the knee is employed to low-pass filter the transmitted shocks. Again, the controller reaction and the load cell data are closely coupled. This setup relies on reducing the controller intervention and therefore allows the natural system dynamics to unfold.

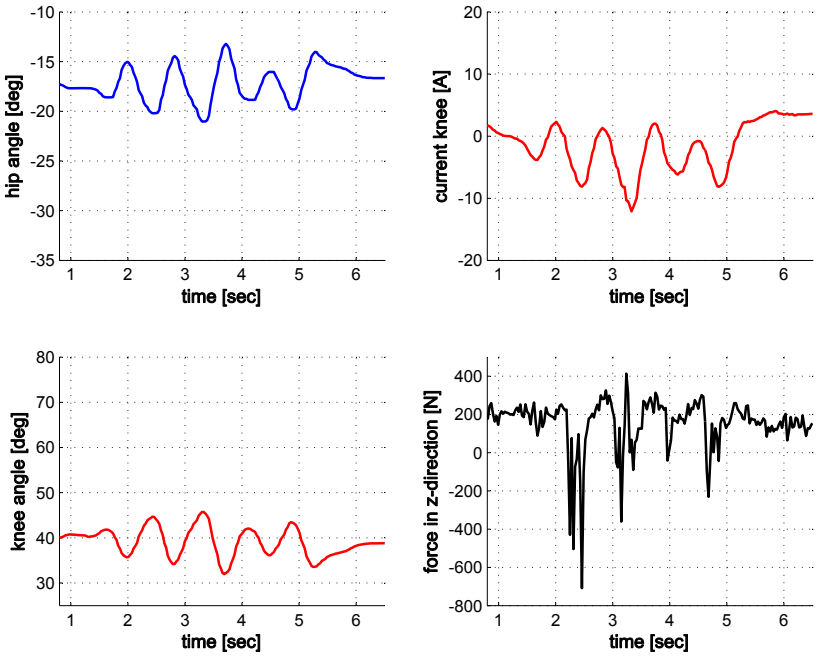


Figure 5.9: Data recorded while the leg setup was exposed to external distortion with a compliance setting of $\iota = 0.5$

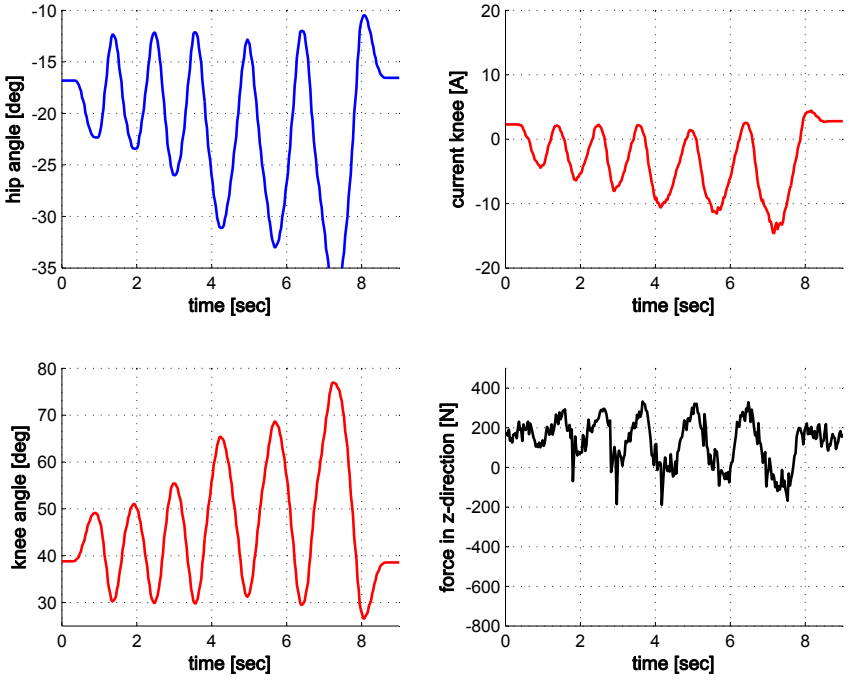


Figure 5.10: Data recorded while the leg setup was exposed to external distortion with a compliance setting of $\iota = 0.3$

5.3.2.4 Push-Off Scenario

Before the influence of different foot constructions on the impact process is investigated, one last experiment concerning the compliant capabilities of the leg is performed. For this purpose the high level control architecture is modified in order to allow for a push-off phase with reduced intensity adjoined by a regular squat phase without the necessity of a lift-off. In order to be able to see the influence of the controller more clearly, only the knee joint is actuated while the hip is entirely relaxed. The result of this process is presented in figure 5.11.

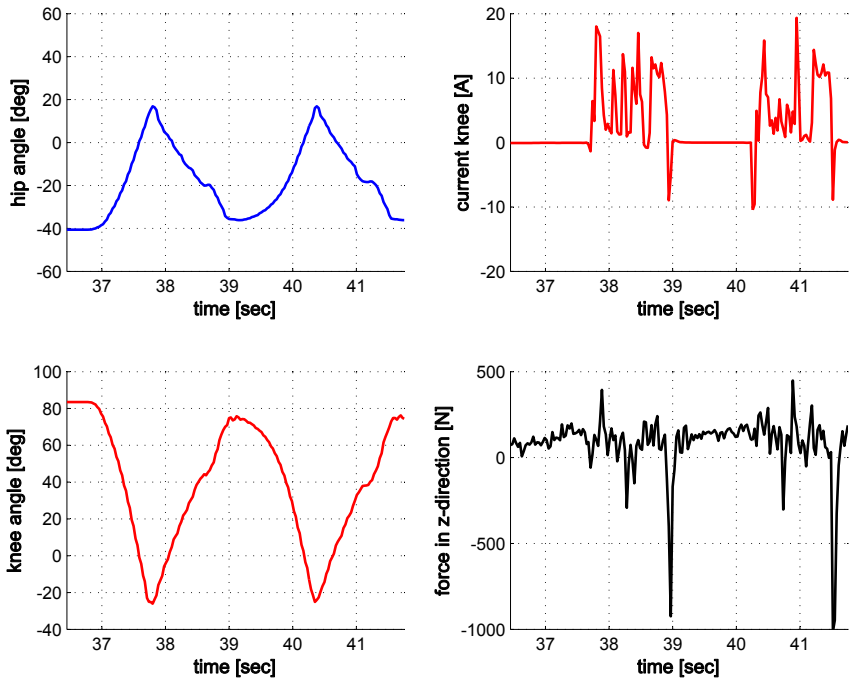


Figure 5.11: Sequence of two cycles with involvement of the push-off and squat reflex

The plot shows two cycles beginning at the lower resting position, i.e. entirely bent knee. At $t = 37$ *sec* the push-off phase is started. At first the lion's share of the momentum is applied by the spring. The fact that the controller is entirely passive shows that the dimensioning of the knee spring seems to be exactly right. Therefore, the controller has to neither apply additional momentum nor is it forced to slow down the natural motion. When the spring is almost entirely contracted, the controller increases the joint torque. Since this phase is utilizing torque control instead of position control, the inertia causes the leg to overshoot the straight position. This is the time for the control instance to shift into the next phase. Thus, the leg is bend again. This is done until the lower 'resting' position is reached at a knee angle of approx. 80 *deg*. This is also visible in the load cell data as a clear peak at $t = 39$ *sec*. Therefore, the next cycle may start.

This experiment is intended to show that the control architecture is basically capable of performing the push-off motion. It reveals however, that the parameters used for the simulation has to be fine tuned in order to achieve a behavior comparable to the one of the simulated leg. Therefore, the conclusion of this experiment has to be that

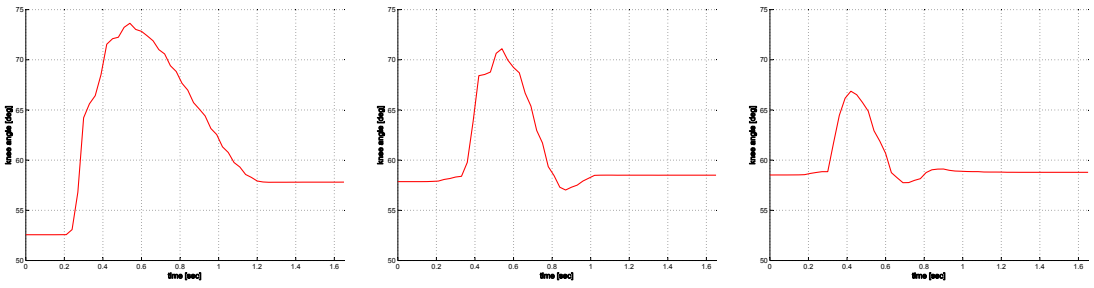
it is basically possible to transfer the parameters from the simulation into the real world but still some effort will be required to adjust them to the new environment.

5.4 Influence of the Foot on the Impact Behavior

One of the most critical moments during a cyclical jump is the impact. This is because the controller has to be able to adapt to the occurring momenta very quickly in order to apply the matching counter momentum. When doing so, timing and amplitude of the applied momentum are the two critical factors. In order to make the task for the controller a lot easier, three different foot setups will be assessed concerning their effectiveness in improving the landing in combination with the joint controller. Thus, in the following paragraphs, the same experiment was performed for all three setups. The data acquired will be discussed at the end of this section.

5.4.1 Setup without Ankle Joint

For the initial experiment, the already described footless setup is used. The point of contact to the ground is formed by a polyurethane hemisphere mounted at the very bottom of the tube equivalent to the tibia bone. The knee is set to an angle of approx. 54 *deg*. The hip joint is relaxed but equipped with an additional mechanical stop to preserve the setup from damage. The leg is lifted up to a height of approx. 11 *cm* above ground and then released. The experiment is repeated for three different joint stiffness settings representing a compliant, a medium, and a rather stiff configuration. Besides the knee angle that is presented in figure 5.12, the force in z-direction, the motor current and the hip angle are evaluated.



(a) $\iota = 0.25$ **(b)** $\iota = 0.50$ **(c)** $\iota = 0.75$

Figure 5.12: Experiment concerning the characteristics during the landing phase

The set joint stiffness increases from left to right. As a result of this the angular amplitude decreases from 21.0 *deg* to 13.2 *deg* and finally 8.4 *deg*. This shows the significant influence of the joint setting on the impact characteristics. In this case the angle almost bisects each time in between the three runs. Moreover an 'overshoot' can be seen in the data for the last two runs. This can be explained with the characteristics of the foot point that is made of a rather bouncy, rubber-like material. With increasing stiffness it is able to absorb more impact energy that causes the leg to slightly bounce off again. In accordance to this, the duration between the impact and a resting position also decreases. While the initial setting of $\iota = 0.25$ takes about 1.02 *sec* to ease up, the most stiff setup manages to do so

within 0.60 *sec*. The motor torque is inversely proportional to this. It increases from an initial amplitude (min. to max.) of 7.5 A to 22.0 A for the last run. Potential changes in the z-force cannot be seen due to a significant noise level in the load cell data.

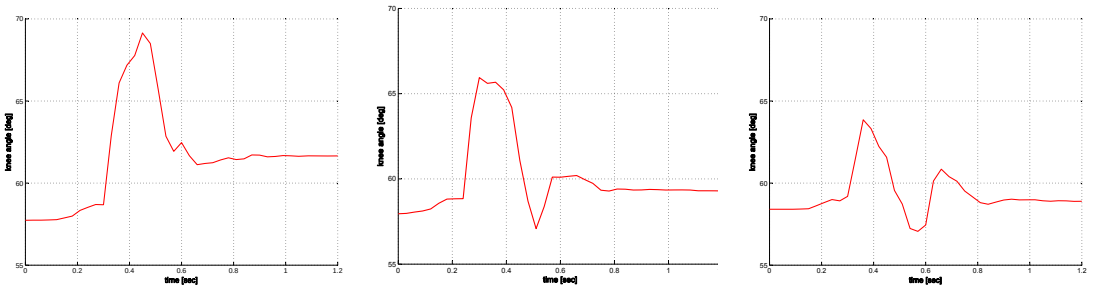
5.4.2 Foot with Series Elastic Element

The second candidate for the foot setup is depicted in figure 5.13. The prototype features a spring that serves as a series elastic element due to the rocker construction. The range of spring travel can be manually adjusted. This allows the user to set a certain pretension. An intermediate setup is used.

Figure 5.13: Mechanical foot construction featuring an elastic element

The experiment is performed the same way as described for the tube structure. The only difference is that the knee angle was now set to approx. 43 *deg* in order to maintain the same foot point relative to the hip pivot axis. The results are depicted in figure 5.14.

In accordance to the last experiment, the angular amplitude decreases from 11.2 *deg* to 5.5 *deg* for increasing stiffness. The rather low level can be explained by the additional rotational DOF introduced by the foot setup. Again the already familiar 'overshoot' occurs. This time however, it is far more severe. The finding can be explained by the increased energy storage potential of the spring in comparison to the PU hemisphere. This assumption is confirmed if the motor current for the stiffness setup depicted in figure 5.18(b) and the load cell data presented in figure 5.19(b) are considered. The time required to compensate for the impact is generally lower than the one observed in the experiment before. It increases with decreasing compliance from 0.48 *sec* to 0.66 *sec* for the most stiff setting. This can be explained with the extra disturbance that is caused by the spring. The overall current amplitude is also



(a) $\iota = 0.25$ **(b)** $\iota = 0.50$ **(c)** $\iota = 0.75$

Figure 5.14: Impact experiment performed with the mechanical foot setup

lower than experienced before (in between 5 A and 14.5 A). This can be explained by the fact that the passively compliant foot absorbs large portions of the initial shock and propagates it back rather slowly. The compliance setting does not seem to have an influence on the recorded force in z-direction once more.

5.4.3 Foot Prosthesis

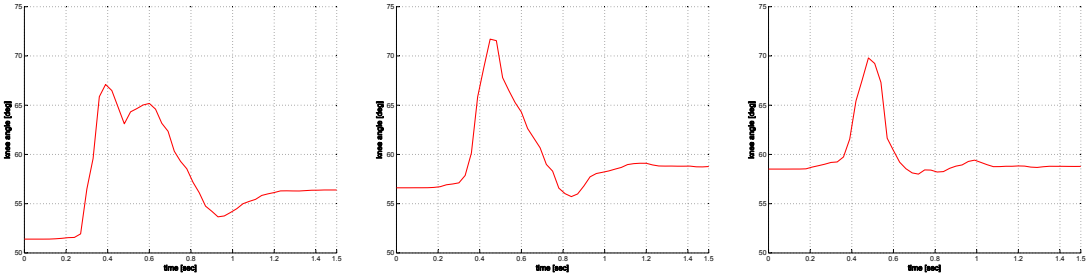
The last potential foot setup that is considered for this series of experiments is a CFRP foot prosthesis made by Otto Bock. The selected model (Trias IC30) is suited for patients up to 40 kg of weight and intended for indoor and outdoor walking. The CFRP spring setup that forms the prosthesis is depicted in figure 5.15.

Figure 5.15: Trias IC30 CFRP foot prosthesis with spring-like properties (source: Otto Bock)

For the experimental setup shown in figure 5.16, a cover made of a synthetical material with damper characteristics is put over the prosthesis.

The prosthesis is mounted below the initial tube structure without the PU foot point. Thus, the lower limb grows a little out of proportion. This causes the lever arm of the knee motor to become one third longer than it used to be for the other two setups. Therefore the torques that have to be applied by the motor have to be put into perspective before they can be compared to the other experiments. The knee angle trajectories for this last setup can be seen in figure 5.17.

Figure 5.16: Experiment setup: prosthesis mounted on lower limb



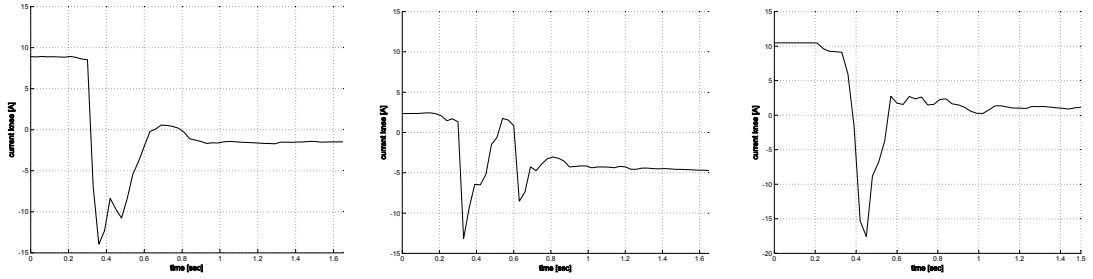
(a) $\iota = 0.25$ **(b)** $\iota = 0.50$ **(c)** $\iota = 0.75$

Figure 5.17: Impact experiment performed with the mechanical foot setup

The decrease in duration and angular amplitude is significantly lower compared to the previous runs. The fastest time of approx $t = 0.90$ *sec* can be achieved for the stiffest setting while the most compliant one results in a delay of 1.11 *sec*. The angular deviation decreases from 15.4 *deg* to 10.5 *deg* while the required current increases from 6.5 A to 28.3 A for the stiffest setup. This can be explained with the excellent energy storage potential of the CFRP springs that transmit the energy batch-wise back into the system. For the first time a significant change in the load cell data can be found. The observed peak forces increase with decreasing compliance.

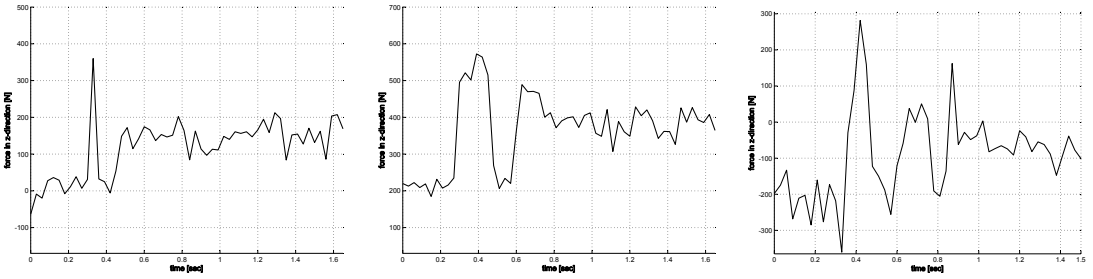
5.4.4 Result Discussion

Now that all three runs of the experiment are presented, it is time to directly compare the results. In order to do so, the controlled motor current (figure 5.18) and the respective load cell data (figure 5.19) for all setups is shown here. In each figure the left graph represents the footless tube construction, the data depicted in the middle was taken from the mechanical foot setup and the one on the right is based on the data gathered during the drop with the foot prosthesis. To increase the expressiveness of the data, all results were recorded during a run with the stiffest controller setting ($\iota = 0.75$). This is because the less energy is transmitted via the parallel knee spring (due to a compliant joint setting), the clearer the results in the controlled torque and the force.



(a) Footless setup **(b)** Mechanical foot **(c)** Prosthesis

Figure 5.18: Momentum applied by the controlled motor for three different mechanical leg configurations. The stiffness was set to $\iota = 0.75$



(a) Footless setup **(b)** Mechanical foot **(c)** Prosthesis

Figure 5.19: Comparison of occurring forces during impact for a controller setting of $\iota = 0.75$

As one can see in figure 5.18, the controller reaction to each setup generates a unique pattern. On the very left (5.18(a)) the result for the tube structure is depicted. As already discussed the impact is rather hard. This can be seen as a clear and narrow peak in the associated load cell data presented in figure 5.19(a). The sudden decline at $t = 0.4\ sec$ is most likely caused by the energy stored in the PU foot point. This assumption is underlined by the occurring forces. After the peak, a force similar to that of the drop phase is recorded before the leg reaches its regular resting characteristics ($t > 0.5\ sec$).

The data gathered during the drop employing the mechanical foot setup with a serial spring is presented in figure 5.18(b) and 5.19(b). The most striking difference in the motor current is the second peak at $t = 0.6\ sec$. In the force data a dilation of the formally narrow peak can be seen. This is most likely caused by the elastic capabilities of the construction. The earlier expressed assumption concerning the second peak is consistent with both the controller and the load cell data. This points to the fact that the pretension of the serial spring has to be matched to the jumping cycle. This is the most important factor in the attempt to preserve the impact energy and feed it back into the system during the push-off phase.

The result of the run featuring the prosthesis is presented in figure 5.18(c) and 5.19(c). Here the most obvious difference to the tube setup is the decaying bouncy pattern visible in the data. It becomes the most obvious in the force data. Peaks of about even distance and declining intensity can be clearly separated from the background noise. The fact that they are almost not visible in the current leads to the conclusion that these shocks are mostly compensated by the elastic elements (prosthesis and knee spring) alone. Concerning the general current level, one has to reconsider the prolonged lever arm. Since torque is defined as the product of the lever arm and the force, the current can be normalized by multiplying it with the lever arm ratio of $\frac{2}{3}$.

In conclusion the result of this set of experiments can be summarized to the following statements:

The tube structure is the easiest to tune in (since the controller setting is alterable) but this comes with the highest stress on the hardware (no series elastic elements). Another major drawback is the absence of energy storage capabilities. This means that with every impact, a lot of energy is lost that has to be refed into the system. This drawback is not occurring with the mechanical foot setup since it features a serial spring. Here, the critical point is the complexity that arises of the combination between the springs and the controller. One has to find a setting that tunes the natural system dynamics in the right way. This is a very elaborate and tedious undertaking. The last candidate is the prosthesis foot. During the experiments it showed the best overall characteristics in combination with the parallel knee spring. Due to its sophisticated elastic properties it allows to reach a resting position quite fast. The negative point about it is that the selection of different prostheses that are suitable for the application scenario is very limited and they are quite pricy.

6. Conclusion and Outlook

6.1 Project Assessment

The task of this thesis was to design and implement a biologically inspired control system for a robot in a highly dynamic application. After investigating the state-of-the-art concerning compliant actuation, control and biological principles associated with this, the concept was composed. Besides the already mentioned features, distribution and natural dynamics were considered.

For this purpose a compliant joint controller was deduced, simulated and implemented on an embedded DSP. This way the first step of a distributed control concept could be realized. Its counterpart, the high level behavior-based control architecture, was designed and implemented on PC using IB2C framework. The already existing leg setup was modified in accordance with the findings of the energy consideration that was performed. For this purpose a parallel spring was included in the new hardware design.

The controller was fiercely tested to determine the overall performance as well as the compliant properties and stability aspects. As a part of this, the most dynamic phase of the jump was investigated in greater detail. In this context three different foot setups were tested concerning their dynamic interaction with the controller.

The overall results seem to be very promising. The desired elastic behavior could be realized using compliant joint controller, parallel springs, and an elastic foot setup. This resulted in natural looking motion patterns. The power consumption could be reduced by making use of the system's natural dynamics. The controller has proven to be very robust concerning external distortions. A good indicator for this is the fact that no adjustments in the parameters had to be performed when the parallel springs were added. One might criticize that no real active compliance [1]

[1] Active compliance would cause the limb controller to actively induce a counter movement if the joint torque grew beyond the desired value. This approach however is only capable of limiting the applied momentum to a desired value. In a hard collision this will result in a simple stop of the forward motion instead of active retraction (reaction of an active compliance controller).

was acquired. For the given scenario however, this is contraindicated since it would result in unpredictable controller reaction in the impact situation.

Unfortunately the final experiment featuring an entire jump cycle could not be performed due to persistent hardware malfunction in the power supply circuit. Therefore one is only able to rely on the simulation data so far.

6.2 Future Work

At the current stage of the project the initial system is ready for use, given that the remaining problem with the power supply can be solved and therefore cyclic jumps can be performed on the real robot. Now it is time to start with the second stage of the work. This means that the influence of the specific parameters of the controller and the architecture has to be investigated on the real setup.

Once this is dealt with it seems very interesting to investigate the capabilities of the distributed concept. For this purpose, delay-critical reflexes with a close coupling to the sensor data should be ported on the DSP in order to be able to explore the impact on the control effort. Furthermore, it will be interesting to incorporate multiple embedded components into the concept, each one responsible for controlling a single joint. This is interesting concerning the reduced wiring overhead and the robustness in case of component failure. Besides, it would be very rewarding to assess the system reaction to external distortions like collision with objects in the environment or uneven ground.

In the long term, the gradual introduction of additional DOF is advised. An additional slider in x-direction or one more rotational joint above the hip, for example, will reduce the mechanical guidance and will make the setup more realistic. Thus, more complex movement patterns can be investigated. This also means that the control architecture has to be extended to be able to deal with the new-won freedom.

A. Table of Abbreviations

AMASC	-	**A**ctuator with **M**echanically **A**djustable **S**eries **C**ompliance
BB	-	**B**ehavior **B**ased
BiMASC	-	**Bi**ped with **M**echanically **A**djustable **S**eries **C**ompliance
CAN	-	**C**ontroller **A**rea **N**etwork
CD	-	**C**oupled **D**rives
CNS	-	**C**entral **N**ervous **S**ystem
COM	-	**C**enter **o**f **M**ass
COP	-	**C**enter **o**f **P**ressure
DC	-	**D**irect **C**urrent
DH	-	**D**enavit **H**artenberg
DOF	-	**D**egree **O**f **F**reedom
DSP	-	**D**igital **S**ignal **P**rocessor
ECD	-	**E**lectric **C**able **D**ifferential
EEG	-	**E**lectro**e**ncephalo**g**raphy
FFT	-	**F**ast **F**ourier **T**ransformation
FIR	-	**F**inite **I**mpulse **R**esponse
FSM	-	**F**nite **S**tate **M**achine
GA	-	**G**enetic **A**lgorithm
GDA	-	**G**ravitationally **D**ecoupled **A**ctuation
HAL	-	**H**ardware **A**bstraction **L**ayer
IB2C	-	**I**ntegrated **B**ehavior-**B**ased **C**ontrol

LTI	-	**L**inear **T**ime **I**nvariant
MACCEPA	-	**M**echanically **A**djustable **C**ompliance and **C**ontrollable **E**quilibriums **P**osition **A**ctuator
PET	-	**P**ositron **E**mission **T**omography
PID	-	**P**roportional-**I**ntegral-**D**erivative
PPAM	-	**P**leated **P**neumatic **A**rtificial **M**uscle
PT1	-	1st order **P**ropor**t**ional delay element
PT2	-	2nd order **P**ropor**t**ional delay element
PU	-	**P**oly**u**rethane
PWM	-	**P**uls **W**idth **M**odulation
PWR	-	**P**ower to **W**eight **R**atio
SEA	-	**S**eries **E**lastic **A**ctuator
SLIP	-	**S**pring **L**oaded **I**nverted **P**endulum
TCP	-	**T**ool **C**enter **P**oint
VC	-	**V**irtual **C**omponent
VIA	-	**V**ariable **I**mpedance **A**ctuation
VMC	-	**V**irtual **M**odel **C**ontrol
VSA	-	**V**ariable **S**tiffness **A**ctuator
WCS	-	**W**orld **C**oordinate **S**ystem
ZMP	-	**Z**ero **M**oment **P**oint

B. Additional Material & Deductions

B.1 Overview of State-Of-The-Art Walking Machines

Robot	**Actuation**	**Gear**	**Control architecture**
Lucy	PPAM	–	trajectory(global), joint(local)
Veronica	MACCEPA	–	central
HRP2-Promet	DC motor	harmonic drive	distributed
LOLA	DC motor	harmonic drive	trajectory (central), smart joints (distr.)
ASIMO	DC motor	harmonic drive	remote controlled, motor (local)
FLAME	DC motor + elastic element		central
SPRING FLAMINGO	SAE	–	central

Robot	**DOF**	**Mass** $[kg]$	**Height**$[cm]$	**Speed** $[km/h]$
Lucy	- / 6 (leg)	30	150	1.4
Veronica	6 leg	5.6	1m	3
HRP2-Promet	6 leg / 30 total	58	154	≈ 2
LOLA	leg / 22 total			2
ASIMO	6 leg / 34 total	54	130	6
FLAME	7 leg	15	130	1.62
SPRING FLAMINGO	6 leg	14	120	2.9

B.2 Deduction of the Parallel Joint Spring Momentum

In order to be able to apply the law of cosines to the problem of finding the angle causing a strain to occur at the spring, one has to determine the angular contributions based on the mounting points. Since both springs are not located on the centerline of the segment relative to the respective adjacent joint pivot point, the offsets cause the angle effecting the spring to be different from the joint angle. Therefore the angle μ can be determined for the upper mounting point (offsets: x_0, y_0) using basic trigonometry:

$$\mu = \tan^{-1}\left(\frac{y_0}{x_0}\right) \tag{B.1}$$

Respectively one receives δ for the lower triangle (offsets: x_1, y_1):

$$\delta = \tan^{-1}\left(\frac{y_1}{x_1}\right) \tag{B.2}$$

Thus, one is able to express the angle effecting the spring β using the measured joint angle α and the two angular contributions found above as :

$$\beta = \alpha - (\mu + \delta) \tag{B.3}$$

In case of the parallel springs, the momentum M_{spring} applied to the actuated joint by the spring can be computed using the spring force F_{spring} in direction of the normal

$$M_{spring} = F_{spring} \cdot y \tag{B.4}$$

with y as the length of the lever arm. The force can be derived applying Hooke's law

$$F_{spring} = \Delta b \cdot k \tag{B.5}$$

with Δb being the spring's change in extension and k the spring constant. Using the law of cosines one receives

$$\Delta b = \sqrt{(x_0^2 + y_0^2) + (x_1^2 + y_1^2) - 2 \cdot \sqrt{x_0^2 + y_0^2} \cdot \sqrt{x_1^2 + y_1^2} \cdot y} - b_0 \tag{B.6}$$

with b_0 being the spring's length if no external forces are applied. Thus one can summarize the equations above as

$$M_{spring} = \cos\beta \cdot k \cdot \sqrt{(x_0^2 + y_0^2) + (x_1^2 + y_1^2) - 2 \cdot \sqrt{x_0^2 + y_0^2} \cdot \sqrt{x_1^2 + y_1^2} \cdot y} - b_0 \cdot y \tag{B.7}$$

B.3 DSP Control Algorithm

Data: Position encoeder data
Result: speed_pos_control

position_actual = ReadEncoder();
position_desired = GetControllerInput(CI_position_desired_edge);
pos_input_error = position_desired - position_actual;
speed_pos_control = pos_input_error · pos_P_gain;

//Limit acceleration & speed;
if *speed_pos_control <> (speed_limit /* ι_{pos}^2 *)* **then**
| speed_pos_control = ± speed_limit ·ι_{pos}^2 ;
end
if *speed_pos_control - speed_pos_control_old <> acc_limit* **then**
| speed_pos_control ±= acc_limit;
end
speed_pos_control = speed_pos_control · ι_{pos};

Algorithm 1: Position Controller

Data: speed_pos_control
Result: torque_speed_control

speed_desired = GetControllerInput(CI_speed_desired_edge) ·
conversion_factor;
speed_actual = (1-α) · position_actual + α · position_actual_old;

//P-Portion;
speed_input_error = speed_desired - speed_actual;
torque_speed_control = speed_input_error · speed_P_gain - speed_actual ·
speed_anticipated_P_gain;

//I-Portion;
speed_integral_value += speed_input_error;
speed_integral_limit = PWM_limit · speed_I_gain · ι_{pos};
if *speed_integral_value <> speed_integral_limit* **then**
| speed_integral_value = speed_integral_limit;
end
torque_speed_control += speed_integral_value · speed_I_gain;

Algorithm 2: Speed Controller

Data: torque_speed_control
Result: PWM_output

torque_desired = GetControllerInput(CI_torque_desired_edge);
torque_sign = TorqueSignHeuristic();
torque_actual = torque_sign · ReadDAConverter() *//Filtered torque measured*;

//Torque Fusion;
torque_speed_control = torque_speed_control *times* ι_{pos}^2;
torque_desired = torque_desired *times* ι_{torque}^2;
tmp = $\iota_{torque} \cdot \iota_{pos}$;
torque_output = (torque_speed_control + torque_desired) / tmp; **if** *torque_output - torque_output_old <> limit* **then**
| torque_output = torque_output_old ± limit;
end

//P-Portion;
torque_input_error = torque_output - torque_actual;
output = torque_input_error · torque_P_gain - torque_actual · torque_anticipated_P_gain;

//I-Portion;
if *torque_output <= limit* **then**
| torque_input_error = torque_input_error ± limit
end
//Reduce drift close to zero;
torque_integral_value += torque_input_error;
torque_integral_limit = PWM_limit · torque_I_gain;
if *torque_integral_value <> torque_integral_limit* **then**
| torque_integral_value = torque_integral_limit;
end
output += torque_integral_value · torque_I_gain;

Output;
if *output <> PWM_limit* **then**
| output = ± PWM_limit;
end
PWM_Output(output);

Algorithm 3: Torque Controller

B.4 Additional Figures

B.4.1 Simulated Disturbance Compensation

In this chapter, the simulation results concerning the ability to adjust to external distortion are presented. For this purpose three typical situations were assessed: The first scenario to be investigated is an unusual early impact. This can occur when e.g. an obstacle is present in the landing zone or the motion is performed on uneven ground. To reproduce this, the platform (that represents the ground level) is raised 20 cm while the leg is in the air. This result in an unexpected early impact during the in-flight phase. The behavior parameters are presented in figure B.1 while the respective sensor data is shown in figure B.2.

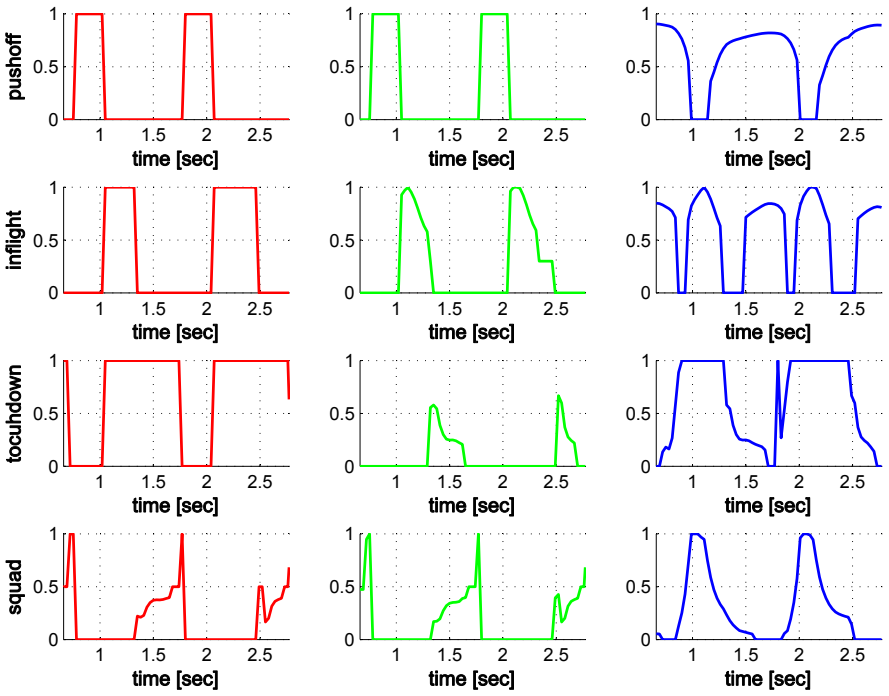


Figure B.1: Behavior data recorded for an too early ground contact

As can be seen, the early impact manifests in an abnormally short in-flight phase between $t = 1.0$ and 1.4. A regular phase can be seen at $t = 2.0$ to 2.5. As visible in the activity data of the in-flight behavior, the impact occurs before it is able to reduce the stiffness to regular level. This results in an unusual hard impact situation which causes a prolonged touchdown phase. It is the logical consequence of the need to compensate for the lower knee bending.

The already mentioned differences can also be observed in the sensor data. Even though the leg configuration is less compliance due to the too early impact, still no significant change in the impact force is noticeable. Extra torque at the knee joint has to be applied to compensate the impact ($t = 1.4$) as opposed to a regular cycle at $t = 2.5$. Thus, it can be stated that the behavior network is able to compensate this kind of unexpected distortions.

The second scenario is intended to imitate the opposing case: an unexpectedly prolonged fall. For this purpose the platform height was increased by 20 *cm*. After

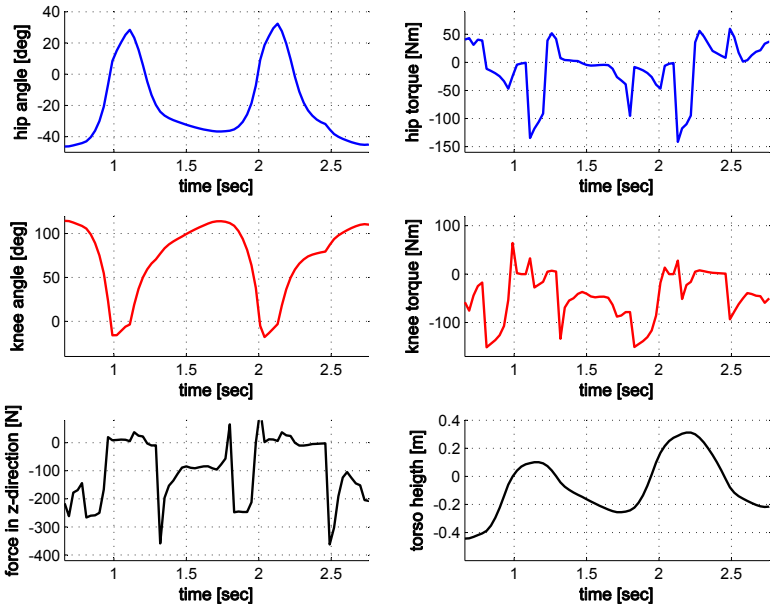


Figure B.2: Sensor data gathered during a premature impact

the jump cycles did stabilized, it was lowered to its original level. The result is a heavier impact due to the increase in kinetic energy. The behavior characteristics are depicted in figure B.3.

The result of the external distortion is showing in a prolonged flight phase ($t = 0.3$ to 0.85 as compared to $t = 1.5$ to 1.95). The adjoining touchdown sequence shows increased activity in order to compensate for the extra impact energy. It is remarkable that no such behavior can be found in the squat reflex. This suggests that the surplus energy is already compensated for by the previous reflex. The respective sensor data is shown in figure B.4.

Expect for the decrease in torso height caused by the lowering platform, the sensor data shows no significant deviation between the disturbed and a regular cycle. Therefore it can be stated that the disturbance can be compensated within one phase.

The last scenario illustrates the occurrence of lateral slip. The platform was moved in x-direction during the attempt to push-off. The behavior data is illustrated in figure B.5.

The slip occurs at the moment of impact. This can be considered as a typical scenario in case the ground does not provide sufficient friction. The result of this is a oscillation between the touchdown and squat reflex around $t = 1.5$. This can be explained by the fact that the slip leads to a rise in angular velocity. Therefore, the touchdown reflex is activated to compensate this. The respective sensor data is presented in figure B.6.

A significant difference in the forces along the z-axis can be observed at ($t = 1.4$ to 1.8). The deviation of the hip level height can be again explained by the activity of the touchdown reflex.

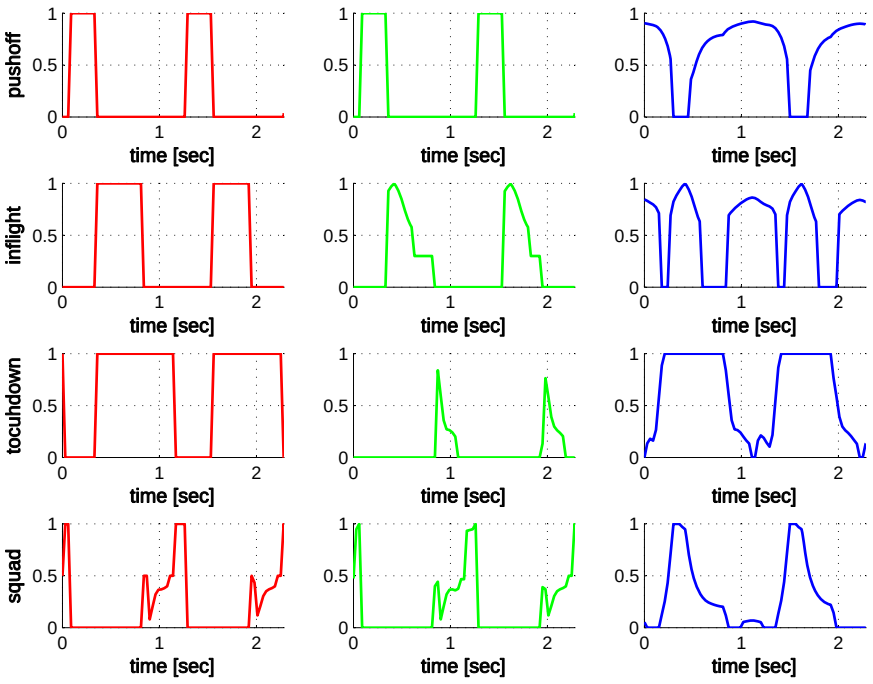


Figure B.3: Behavior data for prolonged fall

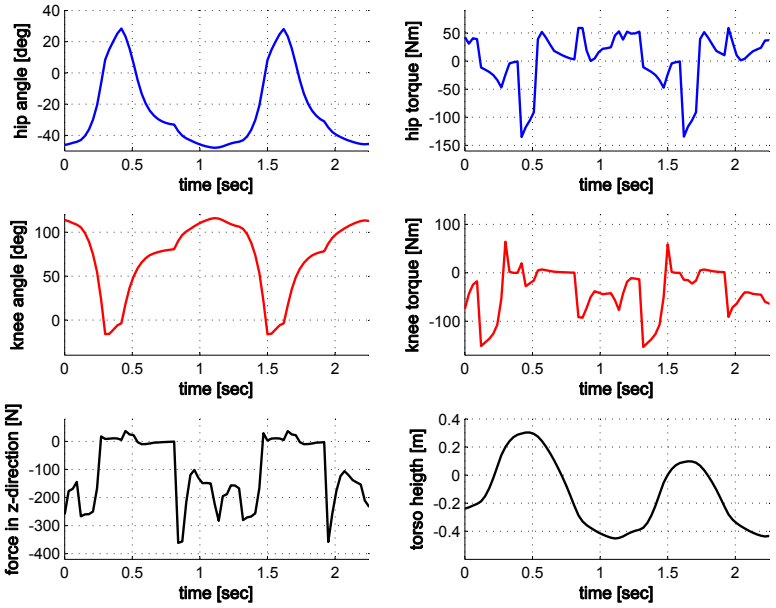


Figure B.4: Sensor readings for too long drop phase

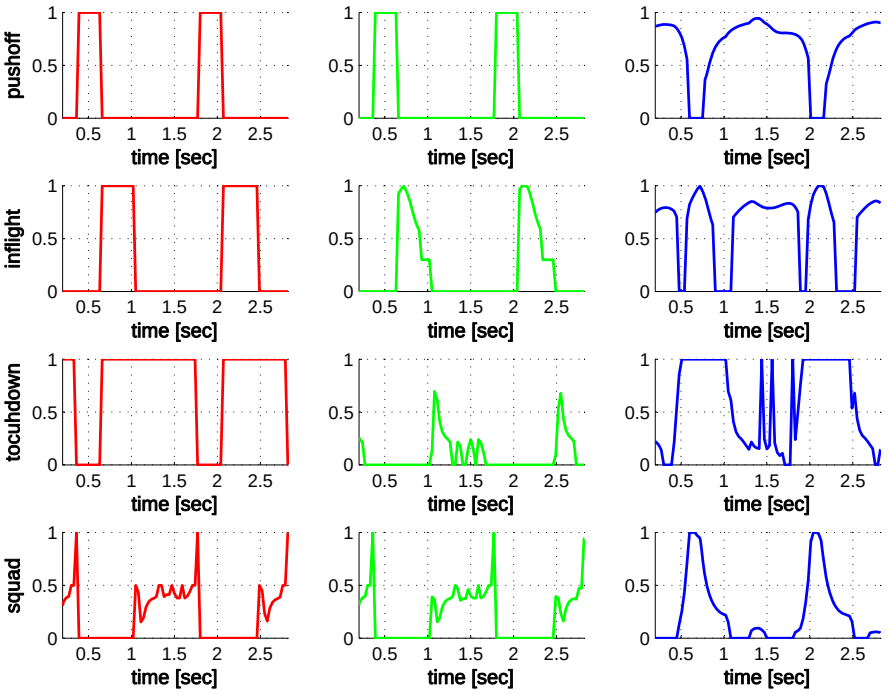


Figure B.5: Impact of lateral slip on the behavior network

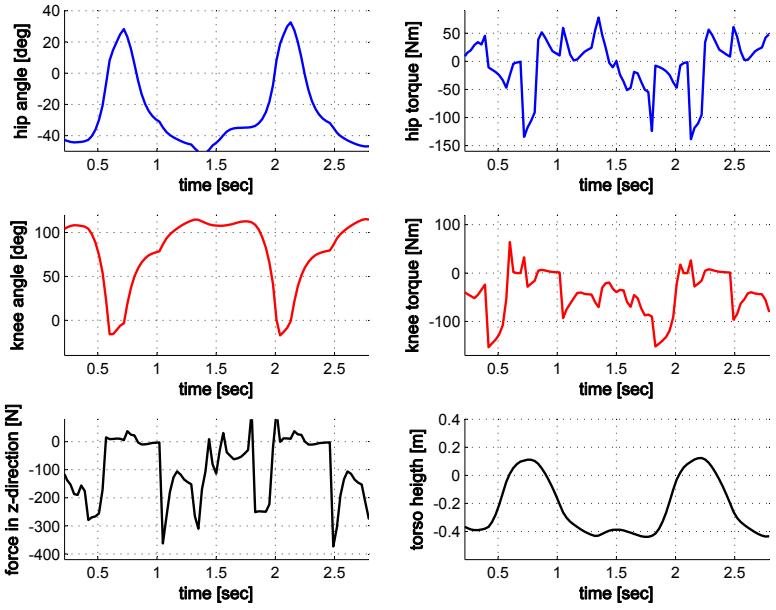


Figure B.6: Sensor data in case of lateral slip during the landing phase

Bibliography

[Albu-Schaeffer 02] A. Albu-Schaeffer, G. Hirzinger, "Cartesian Impedance Control Techniques for Torque Controlled Light-Weight Robots", in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. Washington, DC, USA, May 11–15 2002, pp. 657–663.

[Anderson 05] S.O. Anderson, M. Wisse, C.G. Atkeson, J.K. Hodgins, G.J.Zeglin, B. Moyer, "Powered Bipeds Based on Passive Dynamic Principles", in *Proceedings of the 2005 5th IEEE-RAS International Conference on Humanoid Robotics*. December 5–7 2005, pp. 110–116.

[Bobbert 01] Maarten F. Bobbert, "Dependence of Human Squat Jump Performance on the Series Elastic Compliance of the Triceps Surae: A Simulation Study", *The Journal of Experimental Biology*, 2001.

[Brooks 86] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, April 1986.

[Chatterjee 02] A. Chatterjee, R. Pratap, C. K. Reddy, A. Ruina, "Persistent Passive Hopping and Juggling is Possible Even With Plastic Collisions", *The International Journal of Robotics Research*, vol. 21, no. 7, pp. 621–634, July 2002. DOI: 10.1177/027836402322023213.

[Collins 01] S.H. Collins, M. Wisse, A. Ruina, "A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees", *The international Journal of Robotics Reasearch*, vol. 20, no. 7, pp. 607–615, July 2001.

[Collins 05a] S.H. Collins, A. Ruina, "A Bipedal Walking Robot with Efficient and Human-Like Gait", in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. Barcelona, Spain, April 2005, pp. 1983–1988.

[Collins 05b] S.H. Collins, A.L. Ruina, R. Tedrake, M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers", *Science*, vol. 307, pp. pp. 1082–1085, 2005.

[Curran 08] S. Curran, D.E. Orin, "Evolution of a Jump in an Articualted Leg with Series-Elastic Actuation", in *IEEE International Conference on Robotics and Automation (ICRA)*. Pasadena, CA, USA, May 19–23 2008, pp. 352–358. ISSN 978-1-4244-1647-9.

[Dillmann 04] R. Dillmann, J. Albiez, B. Gaßmann, T. Kerscher, "Biologically Motivated Control of Walking Machines", in *Climbing and Walking Robots - Proceedings of the 7th International Conference CLAWAR 2004*. 2004, pp. 55–69.

[Dillmann 07] R. Dillmann, J. Albiez, B. Gassmann, T. Kerscher, M. Zoellner, "Biologically Inspired Walking Machines: Design, Control and Perception", in *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, F. Pfeiffer, H. Inoue, Eds., London, United Kingdom, January 15 2007, vol. 365, no. 1850, pp. 133 – 151.

[Espenschied 96] K.S. Espenschied, R.D. Quinn, R.D. Beer, H.J. Chiel, "Biologically Based Distributed Control and Local Biologically Based Distributed Control and Local Reflexes Improve Rough Terrain Locomotion in a Hexapod Robot", *Robotics and autonomous systems*, vol. 18, no. 1, pp. pp. 59–64, January 1996. ISSN 0921-8890.

[Foellinger 08] Otto Foellinger, *Regelungstechnik: Einfuehrung in die Methoden und ihre Anwendung*, tenth ed., Huethig Verlag, June 2008. ISBN 978-3778529706.

[Fukashiro 05] S. Fukashiro, T. F. Besier, R. Barrett, J. Cochrane, A. Nagano, D. G. Lloyd, "Direction Control in Standing Horizontal and Vertical Jumps", *International Journal of Sport and Health Sciences*, vol. 3, pp. 1–8, 2005.

[Gevatter 06] Hans-Jürgen Gevatter, Ulrich Grünhaupt, Eds., *Handbuch der Mess- und Automatisierungstechnik in der Produktion*, second ed., ser. VDI-Buch, Springer Verlag, April 2006. ISBN 978-3-540-21207-2.

[Ham 06] R. Van Ham, M. Van Damme, B. Vanderborght, B. Verrelst, D. Lefeber, "MACCEPA, the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator", in *ACTUATOR 2006: The 10th International Conference on New Actuators*. Bremen, Germany, June 14–16 2006.

[Hartenberg 55] R. S. Hartenberg, J. Denavit, "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices", *Journal of Applied Mechanics*, vol. 77, pp. 215–221, 1955.

[Hillenbrand 06] C. Hillenbrand, K. Berns, "Inspection of Surfaces with a Manipulator Mounted on a Climbing Robot", in *37th International Symposium on Robotics (ISR)*. 2006.

[Hurst 04] J.W. Hurst, J. Chestnutt, A. Rizzi, "An Actuator with Physically Variable Stiffness for Highly Dynamic Legged Locomotion", in *Proceedings of the 2004 International Conference on Robotics and Automation (ICRA)*, vol. 5. New Orleans, LA, USA, april 26 – May 1 2004, pp. 4662 – 466.

[Hurst 08] J. W. Hurst, A. A. Rizzi, "Series Compliance for an Efficient Running Gait: Lessons Learned from the Electric Cable Differential Leg", *IEEE Robotics and Automation Magazine*, vol. 15, no. 3, pp. 42–51, September 2008.

[Hyon 02] S.H. Hyon, T. Mita, "Development of a Biologically Inspired Hopping Robot - 'Kenken"', in *Proceedings of the 2002 International Conference on Robotics and Automation (ICRA)*. Washington, D.C., USA, May 2002, pp. 3984 – 3991.

[Kajita 07] S. Kajita, H. Hirukawa, K. Yokoi, K. Harada, *Humanoide Roboter*, S. Kajita, Ed., Berlin, Germany: Akademische Verlagsgesellschaft, AKA GmbH, December 2007. ISBN 978-3-89838-079-9, translation from Japanese into German: K. Ogata.

[Kalveram 91] K. Th. Kalveram, "Controlling the Dynamics of a Two-Joined Arm by Central Patterning and Reflex-Like Processing", *Biological Cybernetics*, vol. 65, no. 1, pp. 65 – 71, May 1991. ISSN 0340-1200.

[Kanehiro 06] F. Kanehiro, Y. Ishiwata, H. Saito, K. Akachi, G. Miyamori, T. Isozumi, K. Kaneko, H. Hirukawa, "Distributed Control System of Humanoid Robots based on Real-time Ethernet", in *IEEE/RSJ International Conference on Intelligent Robots and Systems 2006 (IROS)*. Beijing, China, October 9–15 2006, pp. pp. 2471–2477.

[Kerscher 05] T. Kerscher, J.M. Zoellner, R. Dillmann, G. Stella, G. Caporaletti, "Model and Control of Compliant Joints Driven by Fluidic Muscles", in *ACODUASIS-workshop*. Torino, Italy, October 3 2005.

[Kerscher 07] T. Kerscher, M. Goeller, J. M. Zoellner, R. Dillmann, "Regelungsstrategie fuer Zweibeiniges Elastisches Laufen Mittels 'Virtual Model Control"', in *Autonome Mobile Systeme 2007*, ser. Informatik aktuell, K. Berns, T. Luksch, Eds. Kaiserslautern, Germany: Springer Verlag, October 18–19 2007, pp. 268 –274. ISBN: 978-3-540-74763-5.

[Lim 01] H. Lim, S.A. Setiawan, A. Takanishi, "Balance and Impedance Control for Biped Humanoid Robot Locomotion", in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Maui, Hawaii, USA, October 29 – November 03 2001, pp. 494–499.

[Loeffler 04] K. Loeffler, M. Gienger, F. Pfeiffer, H. Ulbrich, "Sensors and Control Concept of a Biped Robot", *IEEE Transactions on Industrial Electronics*, vol. 51, no. 5, pp. pp 972 – 980, October 2004. ISSN 0278-0046.

[Lohmeier 04] S. Lohmeier, K. Loeffler, M. Gienger, H. Ulbrich, F. Pfeiffer, "Computer System and Control of Biped 'Johnnie"', in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*. New Orleans, LA, USA, April 2004, pp. 4222 – 4227.

[Luhtanen 78] P. Luhtanen, P.V. Komi, "Segmental Contribution to Forces in Vertical Jump", *European Journal of Applied Physiology*, vol. 38, no. 3, pp. pp. 181–188, September 1978. ISSN 1439-6319.

[Luksch 07] T. Luksch, K. Berns, K. Mombaur, G. Schultz, "Using Optimization Techniques for the Design and Control of Fast Bipeds", in *10th International*

Conference on Climbing and Walking Robots (CLAWAR). Singapore, July 16–18 2007.

[Luksch 08] T. Luksch, K. Berns, "Initiating Normal Walking of a Dynamic Biped with a Biologically Motivated Control", in *11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Michines*. Coimbra, Portugal, September 8–10 2008.

[Lunze 07] J. Lunze, *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*, sixth ed., Springer Verlag, February 2007. ISBN 978-3-540-70790-5.

[Matsui 05] T. Matsui, H. Hirukawa, Y. Ishikawa, N.Yamasaki, S. Kagami, F. Kanehiro, H. Saito, T. Inamura, "Distribuited Real-Time Processing for Humanoid Robots", in *The 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005)*. Hong Kong, China, August 17–19 2005, pp. pp. 205–210.

[McGeer 90] T. McGeer, "Passive Dynamic Walking", *International Journal of Robotics Research*, no. 9, pp. 62–82, 1990.

[McGeer 93] T. McGeer, "Dynamics and Control of Bipedal Locomotion", *Journal of Theoretical Biology*, vol. 163, pp. 277–314, 1993.

[Park 01] J.H. Park, "Impedance Control for Biped Robot Locomotion", in *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6. December 6 2001, pp. 870–882.

[Pearson 95] K.G. Pearson, "Proprioceptive Regulation of Locomotion", *Current Opinions in Neurobiology*, vol. 5, no. 6, pp. 768–791, 1995.

[Pfaff 92] G. Pfaff, Ch. Meier, *Regelung elektrischer Antriebe II*, third ed., Oldenbourg Verlag, 1992. ISBN 3-486-22376-3.

[Pfaff 94] G. Pfaff, *Regelung elektrischer Antriebe*, fifth ed., Muenchen, Germany: Oldenbourg Verlag, 1994. ISBN 3-486-23094-8.

[Pfeiffer 07] F. Pfeiffer, H. Inoue, "Walking: Technology and Biology", *Philosophical Transactions of the Royal Society A,*, vol. 365, no. 1850, pp. pp. 3–9, January 15 2007. DOI 10.1098/rsta.2006.1918.

[Pratt 01] J. Pratt, C.M. Chew, A. Torres, P. Dilworth, G. Pratt, "Virtual Model Control: An Intuitive Approach for Bipedal Locomotion", *International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[Pratt 02] J. Pratt, B. Krupp, C. Morse, "Series Elastic Actuators for High Fidelity Force Control", *Industrial Robot Journal*, vol. 29, no. 3, pp. 234–241, 2002.

[Pratt 95] G. A. Pratt, M. M. Williamson, "Series Elastic Actuators", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Pittsburgh, November 1995, pp. 399–406.

[Proetzsch 08] M. Proetzsch, T. Luksch, K. Berns, "Robotic Systems Design Using the Behavior-Based Control Architecture iB2C", *Journal of Robotics and Autonomous Systems*, 2008. submitted to.

[Rizzi 92] Alfred A. Rizzi, Louis L. Whitcomb, Daniel E. Koditschek, "Distributed Real-Time Control of a Spatial Robot Juggler", *IEEE Computer*, vol. 25, no. 5, pp. 12–24, May 1992.

[Robinson 99] D. W. Robinson, J. E. Pratt, D. J. Paluska, G. A. Pratt, "Series Elastic Actuator Development for a Biometric Walking Robot", in *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Machines*. Atlanta, GA, USA, September 19–23 1999, pp. 561 – 568.

[Taylor 94] David G. Taylor, "Nonlinear Control of Electric Machines: An Overview", *Control Systems Magazine, IEEE*, vol. 14, pp. 41–51, December 1994. ISSN: 0272-1708.

[Tonietti 05] G. Tonietti, R. Schiavi, A. Bicchi, "Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction", in *IEEE Int ernational Confonference on Robotics and Automation (ICRA)*. Barcelona, Spain, April 18–22 2005, pp. 528–533.

[Ulbrich 06] Heinz Ulbrich, "Johnnie und Lola", *Zukunft im Brennpunkt*, vol. 5, pp. 29–34, 2006.

[Vanderborght 06] B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, P. Beyl, D. Lefeber, "Torque and Compliance Control of the Pneumatic Artificial Muscles in the Biped 'Lucy"', in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*. Orland, FL, USA, May 2006, pp. 842–847.

[Vanderborght 07] B. Vanderborght, M. Van Damme, R. Van Ham, P. Beyl, D. Lefeber, "A Strategy to Combine Active Trajectory Control with the Exploitation of the Natural Dynamics to Reduce Energy Consumption for Bipedal Robots", in *Proceedings of the IEEE-RAS International Conference on Humanoids Robots - Humanoids 2007*. Pittsburgh, USA, November 29–December 1 2007.

[Wisse 04] M. Wisse, "Essentials of Dynamic Walking, Analysis and Design of Two-Legged Robots", Dissertation, TU Delft, 2004.